UNIVERSITY OF MANITOBA

ON THE ENUMERATION OF ONE-FACTORIZATIONS AND HOWELL DESIGNS USING ORDERLY ALGORITHMS

by

Eric S. T. Seah

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF DOCTOR OF PHILOSOPHY

Department of Computer Science

University of Manitoba

Winnipeg, Manitoba

ISBN 0-315-37239-7

ON THE ENUMERATION OF ONE-FACTORIZATIONS AND HOWELL DESIGNS USING ORDERLY ALGORITHMS

BY

ERIC S.T. SEAH

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

ABSTRACT

In this thesis, we investigate the use of orderly algorithms to enumerate non-isomorphic (perfect) one-factorizations and sets of orthogonal one-factorizations (Howell designs) of regular graphs. These algorithms construct only non-isomorphic one-factorizations, by eliminating isomorphic structures as the one-factorizations are built up from the individual one-factors.

With the help of a high-speed computer, we implement these algorithms for several regular graphs. We enumerate one-factorizations of K_{12} containing prescribed automorphism groups. All perfect one-factorizations of K_{14} containing non-trivial automorphism groups are determined. We complete the census on the one-factorizations and sets of orthogonal one-factorizations for regular graphs of order 10 or less, by performing an enumeration for the graph K_{10} minus a one-factor. We carry out enumerations for 6- and 7-regular graphs on 12 vertices having transitive automorphism groups, and find many new Howell designs. We also study special classes of Howell designs for several graphs on 10, 12 and 14 vertices, such as skew designs, *-designs and **-designs.

Two other algorithms, hill-climbing and backtracking, are used to construct examples of perfect one-factorizations of K_{36} and K_{50}.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. D. R. Stinson. His continuous guidance, advice, encouragement and support over the last few years have made this work a memorable and enlightening experience. I am also indebted to the members of the examining committee, Dr. C. J. Colbourn, Dr. R. S. D. Thomas and Dr. G. H. J. van Rees, for the time which they have spent reading this thesis, and for their many helpful comments and suggestions.

The financial support of the NSERC, the University of Manitoba and the Great-West Life Assurance Company are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Statement of the problem

In this thesis, we study the problems of enumerating *non-isomorphic (perfect) one-factorizations* and *sets of orthogonal one-factorizations (Howell designs)* of regular graphs.

With the assistance of a computer, we used *orderly algorithms* to carry out the enumerations. These algorithms construct only non-isomorphic one-factorizations, by eliminating isomorphic structures as the one-factorizations are built up from the individual one-factors.

The study of one-factorizations belongs to the area known as combinatorial *design* theory. Like many design problems, one-factorizations (of complete graphs, in particular) are closely related to problems such as scheduling round robin tournaments. The importance of the study of one-factorizations cannot be over-emphasized, as illustrated by the following quotations from Mendelsohn and Rosa (see [43]).

"The results of this lead to constructions and applications in other branches of design theory and the recognition of other known designs as special types or orthogonalizations of the basic idea."

"The one-factorization of the complete graph is a building block of resolvable designs and tournament scheduling. As such, it deserves thorough study."

In the following three sections, we define the terminology used in this thesis. We also give a brief description of previous work done in the areas of one-factorizations and Howell designs. In concluding this chapter, we give an overview of the thesis, and some main references.

1.2 Graph theory

A *graph* Gr is defined as an ordered pair (**V**, **E**), where **V** is a finite non-empty set of n elements, and **E** is a finite set of unordered pairs of distinct elements of **V**. The elements of **V** are called *vertices*, and the elements of **E** are called *edges*. We use the set $\{x_0, x_1, ..., x_{n-1}\}$ to denote the n vertices in **V**. The number of vertices in the set **V**, n, is also called the *order* of the graph.

A vertex x is *adjacent* to another vertex y if **E** contains the pair {x, y} (usually called the edge joining x and y). The *degree* of the vertex x is the number of edges incident with it. A graph Gr is *r-regular* if all its vertices have the same degree r. An (n-1)-regular graph on n vertices is known as the *complete graph* of order n, and is usually denoted by K_n. A *complete bipartite* graph Gr on (m+n) vertices, denoted $K_{m,n}$, is a graph where it is possible to partition the vertex set into two subsets, say V_m and V_n ($|V_m| = m$ and $|V_n| = n$), so that every vertex of V_m is adjacent to all vertices of V_n, and no vertex is adjacent to another vertex of its own set. We note that $K_{n,n}$ is n-regular.

A *walk* of the graph Gr is an alternating sequence of vertices and edges in Gr. The sequence begins and ends with a vertex, and each edge in the walk is incident with the vertices immediately preceding and following it; for example,

$\{x_0, \{x_0, x_1\}, x_1, \{x_1, x_2\}, x_2, ..., x_{i-1}, \{x_{i-1}, x_i\}, x_i\}$. To shorten the notation, we will represent the walk by the sequence of vertices, $\{x_0, x_1, x_2, ..., x_{i-1}, x_i\}$, with the understanding that two consecutive vertices in the sequence represent the edge omitted. A *trail* of the graph Gr is a walk such that the edges are all distinct. A *path* of the graph Gr is a trail such that all vertices are distinct, with the possible exception of x_0 and x_i. If $x_0 = x_i$, the path is closed and is called a *cycle*. The *order* of a cycle is defined as the number of vertices in it. A cycle that contains all the vertices of Gr (and hence is of order n) is called a *Hamiltonian* cycle. A 2-regular graph on n vertices, denoted by Q_n, is a collection of one or more vertex-disjoint cycles, the order of each of which is ≥ 3 and $\leq n$.

The n vertices $\{1, ..., n\}$ in the vertex set **V** of a graph Gr may be renamed by some permutation $\alpha \in \mathbf{S}_n$. We write the image of x under α as x^α. Thus $x^\alpha \in \mathbf{V}$ for all $x \in \mathbf{V}$, and we have $\mathbf{V}^\alpha = \mathbf{V}$. The edges in **E** are also renamed under the action of α; that is, $\mathbf{E}^\alpha = \{\{x^\alpha, y^\alpha\} : \{x, y\} \in \mathbf{E}\}$. Note that $\mathbf{E}^\alpha$ may not be identical to **E**. We denote the resulting graph of Gr under the action of α by $Gr^\alpha = (\mathbf{V}, \mathbf{E}^\alpha)$. The *automorphism group* of the graph Gr, denoted Aut(Gr), is the set of permutations such that the resulting graph Gr^α is identical to Gr; that is, $\mathrm{Aut(Gr)} = \{\alpha : Gr^\alpha = Gr, \alpha \in \mathbf{S}_n\}$. A graph Gr is said to have a *transitive* automorphism group if for every vertex $x \in \mathbf{V}$, there exist automorphisms which map x to each vertex of **V**; that is, $\{x^\alpha : \alpha \in \mathrm{Aut(Gr)}\} = \mathbf{V}$ for all $x \in \mathbf{V}$.

1.3 One-factorizations

For an r-regular graph Gr on n vertices, a *one-factorization* (*OF*) of Gr is a partition of the edges in **E** into r *one-factors*, each of which contains n/2 edges that partition the vertices in **V**. It is easy to see that, for an OF to exist, n must be even.

The first literature about OFs of complete graphs, as far as we know, goes back 128 years to the paper of Reiss [48]. However, we cannot rule out the possibility of earlier written sources. The fact that OFs of complete graphs are fairly easy to construct suggests that they may have been considered long before.

It is well-known that there exists an OF of K_{2n} for every positive integer n. In fact, over a hundred years ago, Lucas [42] gave a construction for a class of OFs of K_{2n}, commonly known as the GK_{2n} series. These OFs are constructed as follows.

Let $\mathbf{V} = \mathbf{Z}_{2n-1} \cup \{\infty\}$, and let $f_0 = \{\{j, 2n-j-1\} : j \in \mathbf{Z}_{2n-1} \setminus \{0\}\} \cup \{0, \infty\}$. It is not difficult to see that f_0 is a one-factor. Define $f_i = f_0 + i = \{\{i+j, i+2n-j-1\} : j \in \mathbf{Z}_{2n-1} \setminus \{0\}\} \cup \{i, \infty\}$, for i = 0, 1, ..., 2n-2. Then the set $\{f_0, f_1, ..., f_{2n-2}\}$ is an OF of K_{2n}. Graphically, we can label the vertices of the regular polygon of 2n-1 sides by the elements in **V**, and the centre by ∞. Joining the vertices using the edges of f_0, we obtain a figure such as the one in Figure 1.1 (we use GK_{16} as an example). Rotating the figure successively through an angle of 2π / (2n-1) gives us all the one-factors of GK_{2n}.

Another well-known family of OFs of K_{2n} is the GA_{2n} series (n is odd), which can be constructed from GK_{n+1}, as illustrated by the following example on GA_{10}.

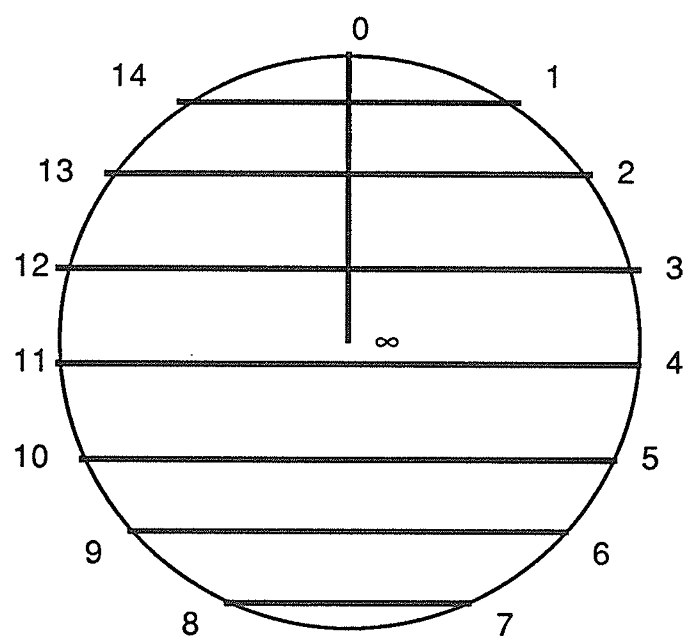
There are two subcollections of one-factors of GA_{10}. We take two distinct copies of GK_6, as represented by the two one-factors $\{\{0, \infty\}, \{1, 4\}, \{2, 3\}\}$ and

{{0', ∞'}, {1', 4'}, {2', 3'}} (see above). Doubling up these two one-factors by combining the two edges {0, ∞} and {0, ∞'} into {0, 0'}, we obtain a one-factor {{0, 0'}, {1, 4}, {2, 3}, {1', 4'}, {2', 3'}}. (Hence the vertex set of GA_{10} is the union of the vertex sets of these two copies of K_6, with the two infinity elements deleted.) Now adding 0, 1, ..., 4 (mod 5) successively to this one-factor, we obtain the first subcollection of one-factors of GA_{10}. The remaining one-factors can be obtained by pairing the vertices of the first K_6 to those of the second one, as follows (using mod 5 arithmetic):

{ {{0, (j+0)'}, {1, (j+1)'}, {2, (j+2)'}, {3, (j+3)'}, {4, (j+4)'}}: j = 1, ..., 4}.

Figure 1.1

The one-factor f_0 of GK_{16}



For an OF F = {f_1, f_2, ..., f_r} of an r-regular graph Gr on n vertices, we denote the resulting OF of F under the action of some permutation $\alpha \in \mathbf{S}_n$ by $F^\alpha = \{f_1^\alpha, f_2^\alpha, ..., f_r^\alpha\}$, where $f_i^\alpha = \{\{x^\alpha, y^\alpha\} : \{x, y\} \in f_i\}$. Two OFs $F = \{f_1, f_2, ..., f_r\}$ and $G = \{g_1, g_2, ..., g_r\}$ of an r-regular graph Gr of order n are *isomorphic* if there exists a permutation α on n vertices such that $F^\alpha = G$; that is, $\{f_1^\alpha, f_2^\alpha, ..., f_r^\alpha\} = \{g_1, g_2, ..., g_r\}$ (note that since F and G are OFs of the same graph Gr, $\alpha \in \text{Aut(Gr)}$). The *automorphism group of F*, denoted by Aut(F), is the set of permutations that fix F; that is, $\text{Aut}(F) = \{\alpha : F^\alpha = F, \alpha \in \text{Aut(Gr)}\}$. We call $\alpha \in \text{Aut}(F)$ an *automorphism* of F.

An OF F of an r-regular graph Gr on n vertices is said to be *cyclic* if Aut(F) contains an automorphism that permutes the vertices of Gr in a single cycle (of length n). An OF F is called 1-*rotational*, if there exists an automorphism in Aut(F) which permutes the r one-factors of F in a single cycle.

Given two distinct one-factors of an OF of an r-regular graph Gr, the 2n edges form a union of disjoint cycles. Furthermore, the order of such a cycle must be an even integer greater than or equal to 4. An OF F of the graph Gr is *perfect* if every pair of distinct one-factors of F forms a Hamiltonian cycle of the graph. We give an example of a perfect OF of K_6, as follows. (In displays, edges will be given without braces.)

1 2	1 3	1 4	1 5	1 6
3 4	2 5	2 6	2 4	2 3
5 6	4 6	3 5	3 6	4 5

It has been conjectured that a perfect OF exists for all K_{2n}. This appears to be a difficult question. In fact, we only know of two infinite families of perfect

OFs: GK_{2n} when 2n-1 is a prime, and GA_{2n} when n is a prime (see [1] and [35]). Perfect OFs were also known to exist on K_{16}, K_{28}, K_{244} and K_{344} (see [2]), and no examples of a perfect OF were known for any other values of n. In this thesis, we shall present two new perfect OFs, of K_{36} and K_{50} (see Chapter 9).

Given a one-factor f of K_{2n}, a *sub-one-factor* f' is a non-empty subset of s edges of f, where $s \leq n$. An OF F of K_{2n} is said to contain a *sub-one-factorization (sub-OF)* of K_{2s}, if there exists a set F' of 2s-1 sub-one-factors from the one-factors of F, such that F' is an OF of a complete graph on 2s vertices. For example, the following OF of K_8 contains a sub-OF of K_4 (on the set of vertices {1, 2, 3, 4}):

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

1.4 Orthogonal one-factorizations and Howell designs

Two OFs F and G of the graph 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, which we usually represent by the set of t integers {1, 2, ..., t}, (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.

Howell designs were first defined by Hung and Mendelsohn in [28], after E. C. Howell, who first constructed such designs (for s = t-1 and t = 4, 6, ..., 30) around 1900 for scheduling bridge tournaments (see [28]).

It is well-known that a pair of orthogonal OFs of Gr, an r-regular graph on 2n vertices, gives rise to a H(r, 2n); and, conversely, the existence of a H(r, 2n) implies the existence of a pair of orthogonal OFs of some r-regular graph on 2n vertices (see [50]). We call Gr the *underlying graph* of the Howell Design. Thus the underlying graph of the following H(4, 6) is K_6 minus a one-factor.

1 6		4 5	2 3
3 4	2 6		1 5
2 5	1 4	3 6	
	3 5	1 2	4 6

We note that the two corresponding orthogonal OFs, F and G, are as follows.

F:

1 2	1 4	1 5	1 6
3 6	2 6	2 3	2 5
4 5	3 5	4 6	3 4

G:

1 2	1 4	1 5	1 6
3 5	2 5	2 6	2 3
4 6	3 6	3 4	4 5

It is easy to see that for H(s, t) to exist, we must have $t/2 \leq s \leq t-1$. In fact, the necessary and sufficient conditions for the existence of a Howell design H(s, t) have been completely resolved, as stated in the following theorems.

Therorem 1.1 ([60], Theorem 6.1)
If s is an odd positive integer and if t is any even integer satisfying the necsssary condition ($t/2 \leq s \leq t-1$), then there is an H(s, t) with precisely three exceptions: there is no H(s, t) for (s, t) = (3, 4), (5, 6), and (5, 8).

Theorem 1.2 ([6], Theorem 6.10)
If s is an even positive integer and if t is any even integer satisfying the necessary condition, then there is an H(s, t) with precisely one exception: there is no H(2, 4).

However, the question as to which regular graphs admit a Howell design appears to be a difficult one. Holyer [26] showed that deciding whether a regular graph Gr admits an OF is NP-complete. The computational complexity of deciding the existence of Howell designs for Gr remains an open problem.

It is well-known that $K_{n,n}$ ($n \neq 2$ or 6), K_{2n} ($n \geq 4$) and K_{2n} - f (where f is a one-factor of K_{2n} and $n \geq 3$) admit Howell designs (see [10], [18] and [19]). Also, everything is known for regular graphs of order up to 10. Aside from these cases, not much else is known in general (see [51]).

We remark that an H(2n-1, 2n), a (2n-1) x (2n-1) square array with K_{2n} as the underlying graph, is commonly referred to as a *Room square* of order 2n

[49]. Nemeth (see [27]) was the first to observe that the existence of a pair of orthogonal OFs of K_{2n} is equivalent to the existence of a Room square of order 2n. The following is an example of a Room square of order 8.

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

This idea of Howell designs can be generalized to higher dimensions, as well. We can define an *i-dimensional Howell design* $H_i(s, t)$ to be an i-dimensional array which satisfies property (1) of the Howell Design, such that each two-dimensional projection of $H_i(s, t)$ is an $H(s, t)$. We refer to an $H_3(s, t)$ as a Howell *cube*. Similar to the 2-dimensional case, an $H_i(r, 2n)$ is equivalent to a set of i mutually orthogonal OFs of the underlying graph Gr, an r-regular graph on 2n vertices.

Given a pair of orthogonal OFs $\{F_1, G_1\}$ of an r-regular graph Gr on n vertices, we say that it is isomorphic to another pair of orthogonal OFs $\{F_2, G_2\}$ of the same graph, if there exists a permutation α on n vertices such that $\{F_1^{\alpha}, G_1^{\alpha}\} = \{F_2, G_2\}$. Thus we define isomorphism of Howell designs in terms of isomorphism of pairs of orthogonal OFs. This can also be generalized to higher dimensions. The isomorphism of sets of i orthogonal OFs (that is, i-dimensional Howell designs), $i \geq 3$, is defined similarly.

Several classes of Howell designs that are of special interest are defined in later Chapters (see Chapters 7 and 8).

1.5 Overview of the thesis

In Chapters 2 to 4, orderly algorithms for enumerating OFs and Howell designs of regular graphs are discussed. Chapter 2 presents orderly algorithms for complete enumeration of OFs of regular graphs, while Chapter 3 gives orderly algorithms that enumerate OFs containing prescribed automorphism groups. Chapter 4 deals with orderly algorithms for enumerating Howell designs.

Chapters 5 to 8 give the results of enumeration of OFs and Howell designs for several graphs. An enumeration of OFs of K_{12} containing certain prescribed automorphism groups is carried out in Chapter 5. A complete enumeration of perfect OFs of K_{14} containing non-trivial automorphism groups is presented in Chapter 6. Chapter 7 investigates OFs and Howell designs of the cocktail-party graph, K_{10} minus a one-factor. In addition to special classes of Howell designs for some graphs on 10 and 14 vertices, OFs and Howell designs of several graphs on 12 vertices are studied in Chapter 8.

Chapter 9 describes the construction of perfect OFs of K_{36} and K_{50} by other algorithms.

Chapter 10 gives a brief summary of this thesis and some open problems.

Appendices 1 to 13 present some of the results of enumerations.

1.6 Main references

For background materials on OFs of complete graphs, we refer the readers to the survey paper by Mendelsohn and Rosa ([43]). A wealth of information on Howell designs can be found in [6], [28], [52], [53] and [60].

For readers interested in orderly algorithms, we recommend the papers by Brown [12] and Read [47].

Many of the results in this thesis can be found in the following papers of Seah and Stinson: [54], [55], [56], [57] and [58]; and the paper of Ihrig, Seah and Stinson [32].

We would like to mention that all of the computer work in this thesis was implemented in PASCAL/VS, and run on the AMDAHL/580 computer at the University of Manitoba.

CHAPTER 2

ORDERLY ALGORITHMS FOR ENUMERATING ONE-FACTORIZATIONS OF REGULAR GRAPHS

2.1 Introduction

One of the most interesting problems related to OFs of r-regular graphs is the enumeration of all pairwise non-isomorphic OFs of these graphs. Many researchers are also interested in OFs that have additional properties (e.g. perfect OFs).

In this chapter, we describe a class of algorithms that can be used to enumerate non-isomorphic OFs of r-regular graphs. These algorithms are known as orderly algorithms, and will be described in detail in the remaining of this chapter. In Chapter 3, we discuss a related class of algorithms, which enumerate OFs containing specified automorphism groups. These are referred to as "automorphism orderly algorithms".

We first record in the following section previous work that has been done with regard to the enumeration of OFs of r-regular graphs.

2.2 Non-isomorphic OFs of regular graphs of small order

Although the existence of OFs of complete graphs K_{2n} for every positive integer n has long been settled, the determination of N(2n), the number of pairwise non-isomorphic OFs of K_{2n}, appears to be difficult. Wallis gave an lower bound on N(2n) and showed that $N(2n) \geq 2$ for $n \geq 4$ [65]. Later, Lindner et al. (see [41]) and Cameron (see [13] and [14]) proved that N(2n) goes to infinity with n. The best result concerning N(2n) that is known today is derived by Cameron in [14], and is as follows:

Theorem 2.1 For sufficiently large non-negative n, $\ln N(2n) \sim 2n^2 \ln 2n$.

In fact, the exact values of N(2n) are only known for a few small values of n.

Theorem 2.2 $N(2) = N(4) = N(6) = 1$; $N(8) = 6$ [17, 22, 66]; $N(10) = 396$ [22].

Enumeration of non-isomorphic OFs of other r-regular graphs on n vertices ($r \neq n-1$) has been carried out by various researchers. Some earlier work was done in [28] and [66]. Rosa and Stinson (see [51]) recently enumerated non-isomorphic OFs of regular graphs of order ≤ 10 and degree ≤ 7. In Chapter 7, we enumerate OFs of K_{10} minus a one-factor.

Denote $N_p(2n)$ to be the number of non-isomorphic perfect OFs of K_{2n}. Not much is known about $N_p(2n)$, except for $n \leq 6$. It is an open question if $N_p(2n) \geq 1$ for all $n \geq 2$.

Theorem 2.3 $N_p(4) = N_p(6) = N_p(8) = N_p(10) = 1$, and $N_p(12) = 5$. [43]

2.3 Comments on algorithms for enumerating one-factorizations of regular graphs

As in many other combinatorial problems, the problem of enumerating the non-ismorphic OFs of r-regular graphs on n vertices quickly becomes computationally intractible when n increases. Although N(8) was first determined by hand [17], the complete enumeration of K_{10} is probably impossible without the use of computers. Since Gelling enumerated N(10) in 1973, computer technology has advanced in leaps and bounds. Yet, the value N(12) still cannot be determined in a reasonable amount of time, which suggests how difficult this problem is (see also Chapter 5).

Due to the lack of success with OFs of complete graphs, many researchers have turned their attention to special classes of OFs. A lot of work has been done recently on perfect OFs of complete graphs (see [2], [3], [25], [29], [30], and [31]). Others have investigated regular graphs of smaller degrees (see for examples, [21] and [51]).

In almost all of these cases, computers have been used to do all or part of the enumeration. Most of these computer algorithms involve first of all constructing all OFs of the graph, followed by the rejection of isomorphic copies. Some of these algorithms do partial isomorphism rejection by means of invariants (see for examples, [21], [22] and [23]). Specific characteristics of the problems on hand are often incorporated into the algorithms to speed up the enumeration process. Thus, the algorithms often cannot be easily modified to apply to other similar problems. Also, this type of approach will probably end up requiring more computer time and storage, when compared to the orderly algorithms discussed in this thesis. More storage is needed because during the process of enumeration, we are dealing with more (partial) structures (some of them are isomorphic); hence additional work (computer time) is required to extend them to (complete) OFs.

Using orderly algorithms, we construct the OFs in a step-by-step, orderly manner. We build up the OFs by adding a one-factor at a time. Every time a one-factor is added, we check to make sure that the (partial) OF we have thus far is not isomorphic to any other one we have already constructed. Thus, throughout the algorithm, we construct only non-isomorphic OFs, by eliminating isomorphic structures as they are being constructed.

In the following sections, we give the definitions and describe the orderly algorithms for enumerating the OFs of K_{2n}. These algorithms can be modified easily for other regular graphs, as discussed in Section 2.7.

2.4 Definitions and orderings for K_{2n}

To explain the orderly algorithms, we need the following definitions.

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, ..., 2n. An edge e will be written as an ordered pair (p, p') with $1 \leq p < p' \leq 2n$. 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) $p_1 < p_2$, or (2) $p_1 = p_2$ and $p_1' < p_2'$.

A one-factor f is written as a set of ordered edges, i.e. $f = (e_1, e_2, e_3, ..., e_n)$, where $e_i < e_j$ whenever $i < j$. 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 $e_{ik} < e_{jk}$.

An OF F of K_{2n} is written as 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 OFs, and f_i, g_i, h_i the corresponding one-factors.

We define an ordering for OFs as follows. For two OFs 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 OF* consisting of an ordered set of i one-factors. We say that i is the *rank* of the partial OF. Note that $F_{2n-1} = F$, a (complete) OF. We can also extend our ordering to partial OFs of rank i, in an analogous manner.

Define $\mathbf{U}_i$ to be the set of all one-factors containing the edge (1, i+1), where i = 1, ..., 2n-1. We say a partial OF $F_i = (f_1, f_2, \ldots, f_i)$ of rank i is *proper* if $f_j \in \mathbf{U}_j$ for all j. We note that a complete OF is proper.

The automorphism group of the complete graph K_{2n} is $\mathbf{S}_{2n}$, the symmetric group on 2n elements. Thus given a proper partial OF F_i (of rank i), we can rename the 2n points using a permutation $\alpha \in \mathbf{S}_{2n}$, and obtain another partial OF (not necessarily proper) of the same graph, denoted F_i^{α}. We say F_i is *canonical* if $F_i^{\alpha} \ge F_i$ for all permutations α. We have the following theorems on canonicity.

Theorem 2.4 If two proper partial OFs of rank i, F_i and G_i, are distinct and are both canonical, then F_i and G_i are non-isomorphic.

Proof By definitions, $F_i^{\alpha} \ge F_i$ and $G_i^{\alpha} \ge G_i$ for all $\alpha \in \mathbf{S}_{2n}$. Without loss of generality, let $F_i < G_i$. If F_i and G_i are isomorphic, then there exists an $\alpha \in \mathbf{S}_{2n}$ such that $G_i^\alpha = F_i$. But then $G_i^\alpha = F_i < G_i$; a contradiction.

Theorem 2.5 If a partial proper OF $F_i = (f_1, f_2, ..., f_i)$ is canonical, and $1 \le j \le i$, then $F_j = (f_1, f_2, ..., f_j)$ is also canonical.

Proof Suppose F_j is not canonical, then there exists an $\alpha \in \mathbf{S}_{2n}$ such that $F_j^\alpha < F_j$. But then $F_j^\alpha \cup \{f_{j+1}, ..., f_i\}^\alpha < F_i$; a contradiction.

Theorem 2.6 If a partial proper OF $F_i = (f_1, f_2, ..., f_i)$ is not canonical, then any complete OF extended from F_i is also not canonical.

Proof Since F_i is not canonical, then there exists an $\alpha \in \mathbf{S}_{2n}$ such that $F_i^\alpha < F_i$. We observe that F_i^α must also be proper. Consequently, if F_i is extended to a complete OF with the set of one-factors $R = \{f_{i+1}, ..., f_{2n-1}\}$, then $F_i^\alpha \cup R^\alpha < F_i \cup R$. Thus, $F_i \cup R$ is not canonical.

By Theorem 2.6, we see that if a proper partial OF F_i is not canonical, then we may discard it. This will reduce the amount of work to be done later.

2.5 Orderly algorithms for enumerating canonical OFs of K_{2n}

We now describe the orderly algorithms that can be used to construct canonical (non-isomorphic) OFs of a complete graph K_{2n}. There are two ways that one can go about generating the OFs of K_{2n}: (1) breadth-first algorithm, and (2) depth-first algorithm.

(1) Breadth-first algorithm

Let $\mathbf{F}_i$ denote the set of canonical proper partial OFs of rank i. A *breadth-first* algorithm generates each set $\mathbf{F}_i$ of canonical proper partial OFs 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 OFs of K_{2n} (in canonical form). The following pseudo-code describes how to generate $\mathbf{F}_{i+1}$ from $\mathbf{F}_i$ (step i+1):

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

FOR each $F_i \in \mathbf{F}_i$ DO

FOR each one-factor $f \in \mathbf{U}_{i+1}$ 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, 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 and proper, so save it for the next step.}

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

(2) Depth-first algorithm

A *depth-first* algorithm uses backtracking. Instead of generating all canonical proper partial OFs of each rank in turn, a depth-first algorithm tries all possible ways of extending each given F_i to an OF, before trying the next F_i. The following recursive pseudo-code describes how to generate from a given F_i, all F_{2n-1} extending F_i, where $0 \le i \le 2n-1$. Let F_0 be the partial OF of rank 0 (an empty set), and $F_0^{\alpha} = F_0$ for all $\alpha \in \mathbf{S}_{2n}$. We invoke the procedure using Depth-first(F_0, 0).

Procedure Depth-first (F_i, i):

IF i = 2n-1 THEN

  F_i is a canonical OF

ELSE

  FOR each $f \in \mathbf{U}_{i+1}$ that is disjoint from each of the 1-factors in 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, discard it and go on to next f;

    {Here $F_i^{\alpha} \cup \{f^{\alpha}\} \ge F_i \cup \{f\}$ for all α. Hence $F_i \cup \{f\}$ is canonical and proper.}

    Depth-first ($F_i \cup \{f\}$, i+1).

It is not difficult to see that both the depth-first and the breadth-first algorithms will enumerate all canonical proper partial OFs of each rank. Since all (complete) OFs are proper, we can determine the number of non-isomorphic OFs by either method.

The algorithms outlined above can easily be modified for certain classes of OFs that are of interest. For example, to construct non-isomorphic perfect OFs of K_{2n}, we modify the algorithms so that a one-factor $f \in \mathbf{U}_{i+1}$ must be disjoint from and form Hamiltonian cycles with each of the one-factors in F_i.

2.6 Canonicity mappings for K_{2n}

In testing whether a proper (partial) OF F_i of K_{2n} is canonical, we can check to see if $F_i^{\alpha} \geq F_i$ for all $\alpha \in \mathbf{S}_{2n}$, the automorphism group of K_{2n}. This is a lot of work, even for small values of n; for example, when 2n = 10, $|\mathbf{S}_{2n}| = 10! = 3628800$.

In practice, we can do a lot better than this. In the case of complete graphs, all one-factors are isomorphic to each other. For any given two one-factors $f_i = ((p_{i1}, p_{i1}'), (p_{i2}, p_{i2}'), \ldots, (p_{in}, p_{in}'))$ and $f_j = ((p_{j1}, p_{j1}'), (p_{j2}, p_{j2}'), \ldots, (p_{jn}, p_{jn}'))$, there exists an α such that $f_i^{\alpha} = f_j$; for example, $\alpha = (p_{i1}\ p_{j1})\ (p_{i1}'\ p_{j1}') \ldots (p_{in}\ p_{jn})\ (p_{in}'\ p_{jn}')$. Thus, the set of proper partial OF of rank 1 consists of only one one-factor, $f_a = ((1, 2), (3, 4), \ldots, (2n\text{-}1, 2n))$, the smallest one-factor of K_{2n}. That is, in using the orderly algorithms, we can start with $\mathbf{F}_1 = \{f_a\}$.

Consequently, we can restrict the canonicity testing to those $\alpha \in \mathbf{S}_{2n}$ such that α maps a one-factor of F_i into f_a (any other α will result in $F_i^{\alpha} > F_i$). Now there are $2^n n!$ ways of mapping one one-factor to another. Therefore, for a proper partial OF F_i, the number of mappings to be carried out equals $i \cdot 2^n n!$, which has a maximum value of $(2n-1)2^n n!$. For example, when 2n = 10, the maximum number of mappings for testing the canonicity of proper partial OF of K_{10} is $9 \cdot 2^5 5! = 34560$, a marked improvement over 3628800.

We can carry this idea one step further. We note that any pair of disjoint one-factors forms a union of disjoint cycles of even lengths (≥ 4). Since any $\alpha \in \mathbf{S}_{2n}$ must preserve the structure of the graph K_{2n}, it follows that a set of two one-factors must map to another pair of disjoint one-factors with the same cycle structure under the permutation $\alpha \in \mathbf{S}_{2n}$. Thus, in testing the canonicity of proper partial OFs of K_{2n}, we can restrict ourselves to those $\alpha \in \mathbf{S}_{2n}$ such that the cycle structure of a pair of disjoint one-factors is preserved. As we shall see in Chapters 5, 6, and 7, this will further reduce the number of mappings that need to be done.

2.7 Enumerating canonical OFs of other regular graphs

For other r-regular graphs Gr on 2n vertices, we can enumerate the non-isomorphic OFs of these graphs by modifying slightly the algorithms for K_{2n} outlined in the preceding sections.

First, we label the 2n vertices by $\{1, \ldots, 2n\}$ in such a way that edges (1, 2), (1, 3), ..., (1, r+1) appear in Gr. An OF F of Gr is written as an ordered set of r one-factors; that is, $F = (f_1, f_2, \ldots, f_r)$. We define $F_i = (f_1, f_2, \ldots, f_i)$ to be a partial OF of rank i, where $1 \leq i \leq r$. A partial OF F_i is proper if $f_j \in \mathbf{U}_j$, where $1 \leq j \leq i$. The orderings are identical to those for complete graphs.

In using the breadth-first algorithm, we would generate $\mathbf{F}_i$ in turn, starting with i = 1 and ending with i = r. Similarly, the recursive algorithm for the depth-first algorithm needs to be changed only so that extension of proper partial OFs stops at i = r.

In general, the order of the automorphism group of an r-regular graph Gr other than K_{2n} is much smaller than that of K_{2n}. Thus it suffices to use the automorphism group of such a graph to carry out the canonicity testing (see Chapter 8); that is, we can use the set $\mathrm{Aut}(Gr) = \{\alpha : Gr^{\alpha} = Gr\}$. We need to consider only mappings $\alpha \in \mathrm{Aut}(Gr)$ because the edge set **E** must be preserved under such α (that is, $\mathbf{E}^{\alpha} = \mathbf{E}$). One advantage of using the set Aut(Gr) is that we use the same mappings α for every partial OF. We remark, however, that if a graph with a "large" automorphism group is to be dealt with (for example, a complete bipartite graph), we may have to use the techniques of mapping pairs of disjoint one-factors to reduce the amount of computer work required.

2.8 Breadth-first versus depth-first algorithms

On the surface, the breadth-first and the depth-first algorithms are not very different. In actual fact, however, they differ in many respects.

Although both algorithms can produce the number of proper partial canonical OFs constructed at each step, the breadth-first algorithm seems to be the more natural way to do it. If the depth-first algorithm does not run to completion, it would not give a complete count of proper partial canonical OFs at the initial steps.

On the other hand, the depth-first algorithm has several advantages over the breadth-first algorithm.

(1) With the depth-first algorithm, we can incorporate *pruning*, by showing that some F_i cannot be extended to an OF, and hence reducing the overall amount of work to be done (and the computer time required).

Pruning can also be incorporated into the breadth-first algorithm, but it cannot be implemented as efficiently as in the case of the depth-first algorithm. The reason is that with the breadth-first algorithm, extension of partial OFs is done on a step-by-step basis. (At step i+1, $\mathbf{F}_i$ is extended to $\mathbf{F}_{i+1}$.) Consequently, we will have to redo the addition and deletion of one-factors at each step (see the next section).

(2) With the depth-first algorithm, no storage is required for the intermediate structures at each step, as compared to the breadth-first method. The storage requirement for the breadth-first algorithm could be quite substantial (for example, refer to Tables 5.2 and 6.2).

(3) The depth-first algorithm is usually faster than the breadth-first algorithm. This is because with the breadth-first algorithm, some calculations have to be redone at the next step (since in general, it is not feasible to store all intermediate results). Thus for example, F_i^α may need to be recalculated at steps i+2, i+3 and so on (see Chapter 6).

2.9 Pruning

Pruning involves showing that certain partial OFs F_i cannot be extended to complete OFs by "looking ahead" into later steps, without actually carrying out the extensions.

We describe the constraints that we used to prune the set of proper partial OFs F_i of K_{2n}, as follows. We remark that this can easily be modified for other regular graphs.

Given an F_i, we define sets $\mathbf{T}_{i+1}, \mathbf{T}_{i+2}, \ldots, \mathbf{T}_{2n-1}$, where $\mathbf{T}_j = \{f \in \mathbf{U}_j : f$ is disjoint from each of the one-factors of $F_i\}$. If any $\mathbf{T}_j = \varnothing$, then F_i cannot be completed to an OF of K_{2n}, so we do not have to investigate any extensions of F_i. For perfect OFs, we require that each of the one-factors in $\mathbf{T}_j$ (where $i+1 \leq j \leq 2n-1$) also forms a Hamiltonian cycle with each of the one-factors of F_i.

Other additional checks can be implemented. For example, in constructing perfect OFs, we observe that if there exist some j such that $|\mathbf{T}_j| = 1$, then the one-factor in $\mathbf{T}_j$ must form a Hamiltonian cycle with at least one one-factor from each of the $\mathbf{T}_k$, $k \neq j$ and $i+1 \leq k \leq 2n-1$.

By eliminating these F_i that cannot be extended to complete OFs, a reduction in computer time generally results. Our experience with the enumeration of perfect OFs of K_{12} indicates that pruning reduces the CPU time required by approximately 50% (see Chapter 6).

2.10 An implementation for pruning

We implemented the pruning scheme described in the previous section with the depth-first algorithm. We note that when a one-factor $f \in \mathbf{U}_i$ is deleted from (or added to) a proper partial canonical OF, we do not have to recompute the sets $\mathbf{T}_{i+1}, \mathbf{T}_{i+2}, \ldots, \mathbf{T}_{2n-1}$. All we need is to be able to dynamically add (delete) one-factors to (from) the sets $\mathbf{T}_{i+1}, \mathbf{T}_{i+2}, \ldots, \mathbf{T}_{2n-1}$, when f is deleted from (added to) the proper partial OF.

In this section, we describe an efficient implementation of dynamic addition and deletion of one-factors.

To facilitate our discussion, we will use a vector notation. We represent the one-factors in $\mathbf{U}_i$ by a vector V_i of one-factors. (Thus $V_i[j]$ refers the jth one-factor in $\mathbf{U}_i$.) Assume $\mathbf{U}_i$ has m_i one-factors. Define two vectors $SOURCE_i$ and $WHERE_i$, each containing m_i elements. $SOURCE_i[j]$ gives the index to the one-factor in V_i, and $WHERE_i[j]$ gives the index to $SOURCE_i$ such that $SOURCE_i[WHERE_i[j]] = j$. $LAST_i$ is defined such that the one-factors given by $V_i[SOURCE_i[j]]$ for $j = 1, ..., LAST_i$ are admissible candidates for extending a proper partial OF.

Initially, we set $LAST_i$ to m_i for $i = 1, ..., 2n-1$ (all one-factors pointed by $SOURCE_i[1]$ to $SOURCE_i[LAST_i]$ are admissible). Thus, $\mathbf{T}_i$ equals $\mathbf{U}_i$ for $i = 1, ..., 2n-1$. We also set $SOURCE_i[j]$ and $WHERE_i[j]$ to j, for $j = 1, ..., m_i$ and $i = 1, ..., 2n-1$. Thus $SOURCE_i[1]$ points to the first one-factor in V_i, and $WHERE_i[1]$ says that the location of the first one-factor in V_i can be found in $SOURCE_i[1]$, etc.

When extending F_i to the next level (step i+1), we need only examine those one-factors in V_{i+1} pointed to by $SOURCE_{i+1}[1]$ through $SOURCE_{i+1}[LAST_{i+1}]]$. If we want to process these one-factors in the same order as in V_{i+1}, then the following pseudo-code could be used:

```
FOR K := 1 to m_{i+1} DO
    IF WHERE_{i+1}[K] <= LAST_{i+1} THEN
        {The Kth one-factor, V_{i+1}[K], is admissible.}
    ELSE
        {The Kth one-factor, V_{i+1}[K], is inadmissible.}
```

If the order in which we process the one-factors is not important, then the following, more efficient pseudo-code could be used:

```
FOR K := 1 to LAST_{i+1} DO
    {The one-factor V_{i+1}[SOURCE_{i+1}[K]] is admissible.}
```

When a one-factor $f \in \mathbf{T}_{i+1}$ is added to F_i, we delete those one-factors in $\mathbf{T}_{i+2}$, ..., $\mathbf{T}_{2n-1}$ that cannot be used (that is, are not admissible) in the further extension of $F_i \cup \{f\}$. The following pseudo-code shows how to delete these one-factors from $\mathbf{T}_j$, for $i+2 \leq j \leq 2n-1$:

```
K := 1;
NO_DELETED_{i+1}[j] := 0;
WHILE (K <= LAST_j) DO BEGIN
    IF V_j[SOURCE_j[K]] is not admissible with respect to F_i ∪ {f} THEN
    BEGIN
        T := SOURCE_j[K];
        W := SOURCE_j[LAST_j];
        SOURCE_j[K] := W;
        SOURCE_j[LAST_j] := T;
        WHERE_j[T] := LAST_j;
        WHERE_j[W] := K;
        LAST_j := LAST_j - 1;
        NO_DELETED_{i+1}[j] := NO_DELETED_{i+1}[j] + 1
    END
    ELSE K := K +1
```

END.

Note that $NO\_DELETED_{i+1}[j]$ gives the number of one-factors deleted from $\mathbf{T}_j$ when a one-factor $f \in \mathbf{T}_{i+1}$ is added to F_i. Also $SOURCE_j[LAST_j + 1]$, $SOURCE_j[LAST_j + 2]$, ..., $SOURCE_j[LAST_j + NO\_DELETED_{i+1}[j]]$ give the indices in V_j of the deleted one-factors.

When we delete the one-factor $f \in \mathbf{U}_{i+1}$ from $F_i \cup \{f\}$, we must add back the one-factors deleted from $\mathbf{T}_{i+2}, \ldots, \mathbf{T}_{2n-1}$ (when f was added to F_i). This can be done easily. The following line of pseudo-code shows the addition of one-factors back to $\mathbf{T}_j$, where $i+2 \leq j \leq 2n-1$:

$$LAST_j := LAST_j + NO\_DELETED_{i+1}[j];$$

We remark that both the dynamic addition and deletion of one-factors are accomplished easily, without having to actually move the one-factors around in the set $\{\mathbf{U}_j : 1 \leq j \leq 2n-1\}$. In fact, addition of one-factors to $\mathbf{T}_j$ takes no time, and deletion of a one-factor requires a constant amount of computer time, regardless of the size of $\mathbf{U}_j$. However, we do need extra storage for the vectors $SOURCE_j$ and $WHERE_j$. For the order of the graphs we are working with, this presents little problem. For example, there are 135135 distinct one-factors for K_{14}, and hence a total of $135135 \cdot 2 \cdot 4$ bytes (or approximately 1080 kilobytes) of extra memory is needed.

CHAPTER 3

ORDERLY ALGORITHMS - ENUMERATING ONE-FACTORIZATIONS OF REGULAR GRAPHS CONTAINING PRESCRIBED AUTOMORPHISM GROUPS

3.1 Introduction

Complete enumeration of (perfect) OFs of complete graphs of relatively small order still remains a difficult problem. Although orderly algorithms as described in Chapter 2 can be used, usually the enumeration cannot be completed within a reasonable amount of time (see Chapter 5), since the number of intermediate (non-isomorphic) structures grows at an astronomical rate. Consequently, many researchers have considered certain special classes of OFs. Anderson investigated starter-induced and even starter-induced OFs of complete graphs K_{2n}, which contain $\mathbf{Z}_{2n-1}$ and $\mathbf{Z}_{2n-2}$ respectively in their automorphism groups (see [2] and also Chapter 9). In [25], Hartman and Rosa enumerated the cyclic OFs of K_{2n} ($n \leq 8$) and showed that a cyclic OF of K_{2n} exists if and only if $n \neq 2^t$, where $t \geq 2$.

In this chapter, we modify the orderly algorithms of Chapter 2 to construct OFs of K_{2n} containing certain prescribed automorphism groups. To distinguish these two classes of algorithms, we call the algorithms in this chapter "automorphism orderly algorithms". We remark that although we refer to complete graphs in the following discussion, these algorithms can easily be modified for other regular graphs.

3.2 Definitions

Orderings on vertices, edges and one-factors are defined as in Chapter 2.

Let A be any subgroup of $\mathbf{S}_{2n}$, the symmetric group on 2n elements. The one-factors of K_{2n} form disjoint orbits under the action of the group A. We are only interested in those orbits which contain edge-disjoint one-factors. We say these are the *eligible* orbits under the action of A.

We order the one-factors in an orbit $O = (f_1, f_2, f_3, \ldots, f_k)$ such that $f_i < f_j$ whenever $i < j$. We say that f_1 is the *representative* of the orbit and write f_1 = rep(O). We define L(O) = k to be the *length* of the orbit O.

We are now ready to define orderings on orbits and OFs. For two orbits O_1 and O_2, we say $O_1 < O_2$ if $\text{rep}(O_1) < \text{rep}(O_2)$. An OF F is written as a list of orbits $(O_1, O_2, \ldots, O_m)$, where $O_i < O_j$ whenever $i < j$. Note that $\Sigma_{1 \le i \le m} L(O_i) = 2n-1$.

A *partial* OF $F_i = (O_1, O_2, \ldots, O_i)$ is written as a list of i orbits. We also define $R = \Sigma_{1 \le j \le i} L(O_j)$ to be the *rank* of F_i. Note that when R = 2n-1, we have a (complete) OF. Also, the number of orbits for two distinct OFs of K_{2n} may be different.

Let $\mathbf{U}_i$ be the set of all one-factors containing the edge (1, i+1). We say that a partial OF $F_i = (O_1, O_2, \ldots, O_i)$ is *proper* if it contains one one-factor from each of $\mathbf{U}_1, \ldots, \mathbf{U}_k$, where $\text{rep}(O_i)$ contains the edge (1, k+1). We note that any (complete) OF is proper, and we have the following theorem:

Theorem 3.1 If $F_i = (O_1, O_2, \ldots, O_i)$ is proper, and $1 \le j \le i$, then $F_j = (O_1, O_2, \ldots, O_j)$ is also proper.

Proof Assume $\text{rep}(O_j) \in \mathbf{U}_m$ and $\text{rep}(O_i) \in \mathbf{U}_k$, where $m \le k$. If F_j is not proper, then these exists a one-factor $f \in \mathbf{U}_l$ which appears in F_i but not in F_j, and $l < m$. Thus f must appear in one of the orbits in $\{O_{j+1}, ..., O_i\}$. But this is impossible, since $f (\in \mathbf{U}_l) < rep(O_j) < rep(O_{j+1})$.

For two proper partial OFs $F_i = (O_1, O_2, ..., O_i)$ and $G_i = (P_1, P_2, ..., P_i)$ that have the same number of orbits i, we say that $F_i < G_i$ if there exists a k, where $1 \le k \le i$, such that $rep(O_l) = rep(P_l)$ for all $l < k$ and $rep(O_k) < rep(P_k)$. Note that we do not require F_i and G_i to have the same rank.

3.3 A-canonicity and quasi-A-canonicity

A proper partial OF F_i is said to be *A-canonical* if $F_i^\alpha \ge F_i$ for all $\alpha \in M(F_i)$, where $M(F_i) = \{\alpha : \alpha \in \mathbf{S}_{2n}$, and α maps any orbit of F_i into an orbit of the same length$\}$.

A proper partial OF F_i that is A-canonical is in general not *canonical* as defined in Chapter 2, since the eligible orbits (one-factors) depend on the prescribed group A. (Some one-factors may not belong to any eligible orbits.)

For example, the A-canonical OF of K_6 containing the automorphism group $A = < \alpha > = < (1)(2)(3\ 4\ 5\ 6) >$ consists of 1 orbit of length 1 and 1 orbit of length 4, as follows:

1 2	3 5	4 6	(orbit of length 1)
1 3	2 4	5 6	(orbit of length 4)
1 4	2 5	3 6	
1 5	2 6	3 4	
1 6	2 3	4 5	

Here, the smallest one-factor $f_a = ((1, 2), (3, 4), (5, 6))$ of K_6 under the action of A is not contained in an eligible orbit, since $f_a^{\alpha} = ((1, 2), (4, 5), (3, 6))$ is not disjoint from f_a. In fact, the *canonical* equivalent of this OF is as follows:

1 2	3 4	5 6
1 3	2 5	4 6
1 4	2 6	3 5
1 5	2 4	3 6
1 6	2 3	4 5

Given an A-canonical OF F, we can determine its canonical form by, for example, mapping all the one-factors of F into the smallest one-factor of K_{2n}, ((1, 2), (3, 4), ..., (2n-1, 2n)). The smallest OF resulting from these mappings is the canonical representation of F.

Similar to canonicity, we have the following theorems on A-canonicity. The proofs are similar to Theorems 2.4 - 2.6.

Theorem 3.2 If two proper partial OFs having the same number of orbits i (not necessary having the same rank), F_i and G_i, are distinct and are both A-canonical, then F_i and G_i are non-isomorphic.

Proof If F_i and G_i are of different ranks, they must be non-isomorphic, since they contain different numbers of one-factors.

Hence, suppose F_i and G_i have the same rank. Without loss of generality, let $F_i < G_i$. If F_i and G_i are isomorphic, then there exists an $\alpha \in M(F_i)$ ($= M(G_i)$) such that $G_i^\alpha = F_i$. By the definitions, $F_i^\alpha \geq F_i$ and $G_i^\alpha \geq G_i$. But then $G_i^\alpha = F_i < G_i$; a contradiction.

Theorem 3.3 If a partial proper OF $F_i = (O_1, O_2, ..., O_i)$ is A-canonical, and $1 \leq j \leq i$, then $F_j = (O_1, O_2, ..., O_j)$ is also A-canonical.

Proof Suppose F_j is not A-canonical; then there exists an $\alpha \in M(F_j)$ such that $F_j^\alpha < F_j$. But then $F_j^\alpha \cup \{O_{j+1}, ..., O_i\}^\alpha < F_i$; a contradiction.

Theorem 3.4 If a partial proper OF $F_i = (O_1, O_2, ..., O_i)$ is not A-canonical, then any complete OF extended from F_i is also not A-canonical.

Proof Since F_i is not A-canonical, then there exists an $\alpha \in M(F_i)$ such that $F_i^\alpha < F_i$. Now F_i^α must also be proper. Consequently, if F_i is extended to a complete OF with the set of orbits $R = \{O_{i+1}, ..., O_r\}$, then $F_i^\alpha \cup R^\alpha < F_i \cup R$. Thus $F_i \cup R$ is not A-canonical.

Let N(A) be the normalizer group of A within $\mathbf{S}_{2n}$; that is, $N(A) = \{\pi : \pi^{-1}A\pi = A, \pi \in \mathbf{S}_{2n}\}$. We remark that $\pi \in N(A)$ maps any eligible orbit into an eligible orbit of the same length, since $(O^{\pi})^A = (O^A)^{\pi} = O^{\pi}$. It should be noted that for a given F_i, $N(A) \leq M(F_i)$, and in general $|N(A)| << |M(F_i)|$.

We say F_i is *quasi-A-canonical* if $F_i^{\alpha} \geq F_i$ for all $\alpha \in N(A)$. A quasi-A-canonical F_i may not be A-canonical, since it is possible to have the situation where all the mappings that take a F_i into its isomorphic copy are not in N(A) (for example, refer to case 20 of Table 5.4).

3.4 Automorphism orderly algorithms

In this section, we outline the automorphism orderly algorithms that can be used to construct OFs of complete graphs containing prescribed automorphism groups.

We can use either the breadth-first or the depth-first algorithms, as described in the following paragraphs (see also Chapter 2). We use N(A) instead of $M(F_i)$ (for a given F_i) to eliminate isomorphic structures, for the following two reasons:

(1) the number of mappings to be performed is significantly reduced;

(2) recalculation of $M(F_i)$ is avoided when F_i changes.

However, the OFs thus generated are not necessarily non-isomorphic. An additional step is therefore required to identify and eliminate the isomorphic copies of these OFs.

(1) Breadth-first algorithm: the following pseudo-code describes how to generate $\mathbf{F}_{i+1}$ from $\mathbf{F}_i$, where $\mathbf{F}_i$ is the set of all quasi-A-canonical proper partial OFs containing i orbits. Note that $\mathbf{F}_0 = \{\emptyset\}$. We repeat the procedure until some $\mathbf{F}_{i+1}$ ($i \geq 0$) is an empty set.

$\mathbb{F}_{i+1} = \varnothing$;

FOR each $F_i \in \mathbb{F}_i$ DO

Determine the smallest integer j such that the edge (1, j+1) is not in F_i;

IF $j \neq 2n+1$ THEN

FOR each orbit O whose representative is in $\mathbf{U}_j$ DO

IF the one-factors of O are disjoint from the one-factors in F_i THEN

FOR each $\pi \in N(A)$ DO

(1) compute O^π and F_i^π;

(2) IF $F_i^\pi \cup \{O^\pi\} < F_i \cup \{O\}$ THEN

$F_i \cup \{O\}$ is not canonical, discard it and go on to next O;

{Here $F_i^\pi \cup \{O^\pi\} \geq F_i \cup \{O\}$ for all π, save $F_i \cup \{O\}$ for the next step.}

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

ELSE

F_i is a complete OF.

(2) Depth-first algorithm: the following recursive pseudo-code outlines how to generate from a given F_i, all quasi-A-canonical OFs extending F_i:

PROCEDURE Depth-first (F_i)

Determine the smallest integer j such that the edge (1, j+1) is not in F_i;
IF j = 2n+1 THEN
    F_i is a quasi-A-canonical OF;
ELSE
    FOR each orbit O whose representative is in $\mathbf{U}_j$ DO
        IF the one-factors of O are disjoint from the one-factors of F_i THEN
            IF $F_i^\pi \cup \{O^\pi\} \geq F_i \cup \{O\}$ for all $\pi \in N(A)$ THEN
                Depth-first ($F_i \cup \{O\}$).

The comments in Chapter 2 about the differences between the breadth-first and the depth-first algorithms also apply here.

We remark that when A is the trivial group of order one, all one-factors are eligible orbits of length one and $N(A) = \mathbf{S}_{2n}$. Consequently, these two algorithms reduce to the orderly algorithms described in Chapter 2. In this case, we would obtain a complete enumeration of the OFs of K_{2n}.

Note that the algorithms above can be easily modified for subclasses of OFs that may be of interest. For example, to enumerate perfect OFs, we modify the algorithms so that pairs of distinct one-factors are both disjoint and Hamiltonian.

Similar to the orderly algorithms in Chapter 2, automorphism orderly algorithms can be modified easily for other r-regular graphs Gr on 2n vertices.

Again, we would label the 2n vertices by {1, ..., 2n} in such a way that edges (1, 2), (1, 3), ..., (1, r+1) appear in Gr. Orderings are similar to complete graphs, and complete OFs would have rank = r. The '2n+1' in the pseudo-code for the depth-first and the breadth-first algorithms in the previous section would be changed to 'r+1'.

3.5 Canonicity testing

In some cases, the number of mappings of N(A) required for quasi-A-canonicity testing, where A is the prescribed automorphism group, could be so large that the enumeration probably cannot be done in a reasonable amount of time (for example, see Section 6.4). This occurs when the order of A is "small". Consequently, we can use one of the following three strategies:

(i) Omit the canonicity testing of partial structures entirely;

(ii) Carry out "partial" canonicity testing of partial structures: use a proper subset of N(A);

(iii) Omit canonicity testing for certain steps. It should be noted that this can also be implemented for the depth-first algorithm, although it is more natural with the breadth-first algorithm.

In general, these strategies are only useful in situations when a small number of OFs are expected. Otherwise, the number of isomorphic copies may explode and much work would be required later. Examples illustrating the use of these strategies include the enumeration of perfect OFs that contain certain automorphism groups, and the enumeration of some special classes of OFs which we suspect to be non-existent (see Section 6.4).

3.6 Pruning

As in the case of orderly algorithms in Chapter 2, pruning can be incorporated into automorphism orderly algorithms. The main difference is that we now prune the orbits, instead of the one-factors.

To implement pruning with automorphism orderly algorithms for complete graphs K_{2n}, we keep a list of orbit representatives of all the eligible orbits under the action of prescribed automorphism group A. Let $\mathbf{O}_i$ be the set of orbits whose orbit representatives contain edge (1, i+1). Then given a proper partial OF $F_k = \{O_1, O_2, ..., O_k\}$, where $rep(O_k)$ contains the edge (1, i+1), we can determine the sets $\mathbf{P}_{i+1}, \mathbf{P}_{i+2}, ..., \mathbf{P}_{2n-1}$, where $\mathbf{P}_j$ is a subset of $\mathbf{O}_j$, and every orbit of F_k is disjoint from every orbit in $\mathbf{P}_j$, for $i+1 \leq j \leq 2n-1$.

We can dynamically delete orbits from (or add orbits to) $\mathbf{P}_j$, where $i+2 \leq j \leq 2n-1$, when an orbit is added to (or deleted from) F_i. The process involved is very similar to the scheme described in Sections 2.9 and 2.10.

To check whether a proper partial OF F_i can be extended to complete OFs, we need to do some additional work. Essentially, we need to determine W = $\{j : (1, j+1) \in f$, and f is a one-factor of $F_i\}$, and Y = $\{j : (1, j+1) \in f$, and f is a one-factor in an orbit of $\mathbf{P}_k$, $i+1 \leq k \leq 2n-1\}$. Now if $W \cup Y \neq \{1, ..., 2n-1\}$, the partial OF F_i cannot be extended to complete OF.

CHAPTER 4

ORDERLY ALGORITHMS FOR ENUMERATING HOWELL DESIGNS

4.1 Introduction

Enumeration of orthogonal OFs (that is, Howell designs) for regular graphs has been carried out by various researchers. Define $N_i(2n)$ to be the number of non-isomorphic sets of i mutually orthogonal OFs (or i-dimensional Howell designs) of K_{2n}. Beaman showed that $N_2(10) = 257630$ [9]. In [7] and [21], it was proved that $N_3(10) = 267$, $N_4(10) = 1$ (and $N_5(10) = 0$). In [51], Rosa and Stinson also enumerated Howell designs of regular graphs of order ≤ 10 and degree ≤ 7.

In this Chapter, we extend the canonicity concept for OFs in Chapter 2 to orthogonal OFs, and devise orderly algorithms that can be used to enumerate non-isomorphic orthogonal OFs of regular graphs.

4.2 Definitions

As before, we will give the definitions and algorithms in terms of complete graphs K_{2n}. Generalization to other regular graphs is easy and will be dealt with at the end of this chapter.

The orderings of vertices, edges, one-factors and OFs are identical to Chapter 2.

We write a set of two orthogonal OFs F and G as an ordered pair (F, G), with F < G. Denote $F = (f_1, f_2, ..., f_{2n-1})$, $G = (g_1, g_2, ..., g_{2n-1})$. Given two Howell designs (F_1, G_1) and (F_2, G_2) having the same underlying graph, we define $(F_1, G_1) < (F_2, G_2)$ if either (1) $F_1 < F_2$, or (2) $F_1 = F_2$ and $G_1 < G_2$.

We extend the canonicity concept in Chapter 2, and say that (F, G) is *canonical* if, for all $\alpha \in \mathbf{S}_{2n}$, $(F, G)^\alpha \geq (F, G)$. We have the following theorems.

Theorem 4.1 If (F, G) is canonical, then F must be canonical.

Proof If F is not canonical, then there exists an $\alpha \in \mathbf{S}_{2n}$ such that $F^\alpha < F$. But then $F^\alpha < F < G$, and hence $(F, G)^\alpha < (F, G)$; a contradiction.

Theorem 4.2 If (F_1, G_1) and (F_2, G_2) are both distinct and canonical, then they are non-isomorphic.

Proof Without loss of generality, let $(F_1, G_1) < (F_2, G_2)$.
Suppose (F_1, G_1) and (F_2, G_2) are isomorphic, then there exists an $\alpha \in \mathbf{S}_{2n}$ such that $(F_1, G_1) = (F_2, G_2)^\alpha$. Since (F_2, G_2) is canonical, we have $(F_2, G_2)^\alpha \geq (F_2, G_2)$. But then $(F_1, G_1) = (F_2, G_2)^\alpha \geq (F_2, G_2)$; a contradiction.

It follows from Theorem 4.1 that to construct the Howell designs of complete graph K_{2n}, we can start with the set of canonical OFs $\mathbf{F}_{2n-1}$ of K_{2n}, and generate all OFs G that are orthogonal to and greater than F for each $F \in \mathbf{F}_{2n-1}$. It is easy to see that a given (F, G), where F < G and F is canonical, is not necessarily canonical. Theorem 4.2 suggests that we can apply canonicity testing to all (F, G) pairs generated and eliminate the non-canonical (isomorphic) ones.

4.3 Canonicity mappings

As in the case of constructing canonical OFs of K_{2n}, using the automorphism group of K_{2n}, $\mathbf{S}_{2n}$, to carry out the canonicity testing is generally unacceptable.

In fact, in testing whether (F, G) is canonical, it suffices to check all $\alpha \in \mathbf{S}_{2n}$ such that either F^α or G^α is canonical (by Theorem 4.1). That is, we can ignore those $\alpha \in \mathbf{S}_{2n}$ such that $F^\alpha > F$ and $G^\alpha > G$, for then we have $(F, G)^\alpha > (F, G)$.

For $\alpha \in \mathbf{S}_{2n}$ that makes F^α canonical, we must have $F^\alpha = F$, 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.

For $\alpha \in \mathbf{S}_{2n}$ that makes G^α canonical, we can map each of the one-factors of G into the smallest one-factor of K_{2n} (which must necessarily be a one-factor of F), namely, $f_a = ((1, 2), (3, 4), ..., (2n-1, 2n))$. As discussed in Chapter 2, the number of such α equals $(2n-1)2^n n!$, which is still a lot of work. Applying the idea of mapping a pair of distinct one-factors to another pair as described in Chapter 2, we can cut down substantially the number of α required. In this case, it suffices to map every pair of distinct one-factors of G to the smallest pair of distinct one-factors of F (that is, f_a and the one-factor containing the edge (1, 3)), for otherwise, $G^\alpha > F$. This is the approach we use (see Chapters 7 and 8). 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.

4.4 An orderly algorithm for Howell designs

We now outline the algorithm that we use to generate all the non-isomorphic Howell designs for K_{2n}:

FOR each $F \in \mathbf{F}$ of canonical OFs of K_{2n} DO:

1. Generate from $\mathbf{U}_i$, i = 1, ..., 2n-1 the set **T** of one-factors that intersect each of the one-factors of F in at most one edge.
2. Construct all possible OFs G, which consist only of one-factors from **T**, discarding those G's < F. These G's are all orthogonal to F. Let G = $(g_1, g_2, ..., g_{2n-1})$.
3. IF no G's were constructed in step 2, go on to next F.
4. Determine the automorphism group B of F; that is, $B = \{\alpha : F^\alpha = F\}$.
5. FOR each G DO:
   (a) IF there exists some $\alpha \in B$ such that $G^\alpha < G$, (F, G) is not canonical, go to next G. Otherwise proceed to (b).

(b) Determine $C = \{\alpha : (g_i, g_j)^\alpha = (f_1, f_2)$, for $i \neq j$, $g_i, g_j \in G$; $f_1, f_2 \in F\}$. IF $(F, G)^\alpha \geq (F, G)$ for all $\alpha \in C$, (F, G) is canonical; otherwise (F, G) is not canonical.

4.5 Higher dimensional Howell designs

We write a set of i mutually orthogonal OFs of K_{2n} (which corresponds to an i-dimensional Howell design) as an ordered i-tuple $(F_1, F_2, ..., F_i)$ with $F_j < F_k$ whenever $j < k$. We say that $(F_1, F_2, ..., F_i)$ is *canonical* if $(F_1, F_2, ..., F_i)^\alpha \geq (F_1, F_2, ..., F_i)$ for all $\alpha \in \mathbf{S}_{2n}$. We have the following theorems, which are generalizations of Theorems 4.1 and 4.2.

Theorem 4.3 If $(F_1, F_2, ..., F_i)$ is canonical, then for $j = 1, ..., i-1$, $(F_1, F_2, ..., F_j)$ is canonical.

Proof Assume for some j, where $1 \leq j \leq i-1$, $(F_1, F_2, ..., F_j)$ is not canonical. Then there exists an $\alpha \in \mathbf{S}_{2n}$ such that $(F_1, F_2, ..., F_j)^\alpha < (F_1, F_2, ..., F_j)$. Now since $(F_1, F_2, ..., F_i)$ is canonical, we have $(F_1, F_2, ..., F_i)^\alpha \geq (F_1, F_2, ..., F_i)$, or $(F_1, F_2, ..., F_j)^\alpha \cup (F_{j+1}, F_{j+2}, ..., F_i)^\alpha \geq (F_1, F_2, ..., F_j) \cup (F_{j+1}, F_{j+2}, ..., F_j)$, which is impossible because $(F_1, F_2, ..., F_j) < (F_{j+1}, F_{j+2}, ..., F_i)$ and $(F_1, F_2, ..., F_j)^\alpha < (F_1, F_2, ..., F_j)$.

Theorem 4.4 If $(F_1, F_2, ..., F_i)$ and $(G_1, G_2, ..., G_i)$ are both distinct and canonical, then they are non-isomorphic.

Proof Without loss of generality, assume $(F_1, F_2, ..., F_i) <$

$(G_1, G_2, ..., G_i)$.

If $(F_1, F_2, ..., F_i)$ and $(G_1, G_2, ..., G_i)$ are isomorphic, then there exist an α such that $(F_1, F_2, ..., F_i) = (G_1, G_2, ..., G_i)^\alpha$. Since $(G_1, G_2, ..., G_i)$ is canonical, we have $(G_1, G_2, ..., G_i)^\alpha \geq (G_1, G_2, ..., G_i)$. But then $(F_1, F_2, ..., F_i) = (G_1, G_2, ..., G_i)^\alpha \geq (G_1, G_2, ..., G_i)$; a contradiction.

4.6 An orderly algorithm for Howell cubes

By Theorem 4.3, we know that for a set of three mutually orthogonal OFs (a Howell cube), (F, G, H), to be canonical, F must be canonical, so must (F, G). These observations suggest that the following algorithm can be used to construct the Howell cubes:

FOR each non-isomorphic F of K_{2n} DO:

1. Construct from **T** the set of all OFs, **G** = {G : F < G and G is orthogonal to F}, as in steps 1 and 2 of the algorithm for Howell designs (Section 4.4).
2. Examine all pairs of OFs G and H, where G < H and G, H ∈ **G**. If G and H are orthogonal, then we have a set (F, G, H) of three mutually orthogonal OFs.
3. Determine which triples (F, G, H) are canonical.

In determining the canonicity of the set of {(F, G, H)} (step 3 above), we can make use of Theorem 4.3 to first of all eliminate those (F, G, H) of which

(F, G) is not canonical. For the remaining triples, we can restrict the mappings to those $\alpha \in \mathbf{S}_{2n}$ that makes F^{α}, G^{α} or H^{α} canonical. (The idea is very similar to the canonicity testing of 2-dimensional Howell designs discussed in Sections 4.3 and 4.4.)

4.7 An orderly algorithm for higher dimensional Howell designs

We present in this section an algorithm to construct all non-isomorphic (i+1)-dimensional Howell designs for a given set of all canonical non-isomorphic i-dimensional Howell designs $\mathbf{H}_i$.

For each $H = (F_1, F_2, \ldots, F_i) \in \mathbf{H}_i$ do

1. Generate from $\mathbf{U}_j$ $(1 \leq j \leq 2n-1)$ the set $\mathbf{T}$ of one-factors that intersect each of the one-factors of F_k $(1 \leq k \leq i)$ in at most one edge.
2. Construct the set $\mathbf{F} = \{F\}$ of all possible OFs, which consist only of one-factors from $\mathbf{T}$, discarding those $F < F_i$.
3. For each $F \in \mathbf{F}$, $H \cup \{F\}$ is an (i+1)-dimensional Howell design.

We note that the set of (i+1)-dimensional Howell designs produced by the algorithm above are not necessarily canonical (non-isomorphic). Thus, as in the cases of Howell designs and Howell cubes, we need to eliminate isomorphic copies (see Sections 4.3 and 4.6).

4.8 Other regular graphs

The algorithms described in the preceding sections can be modified easily for other r-regular graphs Gr.

With the vertices labelled and OFs constructed as described in Section 2.7, the Howell designs can be obtained using the algorithm in Section 4.4; the only modification required is to change the '2n-1' in line 1 of the algorithm to 'r'. For higher dimensional Howell designs, algorithms described in Sections 4.6 and 4.7 can be used.

We would like to add that the same remarks in Section 2.7 about the canonicity testing for r-regular graphs also apply here: the full automorphism group of Gr is usually used since its order is generally fairly small (see Chapter 8).

CHAPTER 5

ENUMERATING ONE-FACTORIZATIONS OF K_{10} AND K_{12}

5.1 Introduction

In 1973, Gelling [22] enumerated the non-isomorphic OFs of K_{10} with the assistance of the computer. Since then, the number of non-isomorphic OFs of the complete graph of the next higher order, K_{12}, still remains to be settled.

In this chapter, we describe how the orderly algorithms described in Chapter 2 can be used to construct the non-isomorphic OFs of K_{10}. We also show how automorphism orderly algorithms in Chapter 3 are used to enumerate non-isomorphic OFs of K_{12} containing prescribed automorphism groups, and obtain the following lower bound.

Theorem 5.1 For the complete graph K_{12}, excluding those OFs containing exactly one automorphism of six disjoint cycles of length two, there are precisely 56391 non-isomorphic OFs with non-trivial automorphism groups.

5.2 One-factorizations of K_{10}

In using the orderly algorithms in Chapter 2, we note that there are different ways one can carry out the canonicity testing (see Section 2.6). The approach we used is to map a pair of one-factors of $F_i \cup \{f\}$ into a *fixed* pair of one-factors.

We observe that any two disjoint one-factors of K_{10} form either two disjoint cycles of lengths 4 and 6 (type '46') or a Hamiltonian cycle of length 10 (type '10'). The smallest one-factor in $\mathbf{U}_2$ 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 one-factor in $\mathbf{U}_2$ that forms a type '10' with f_a is f_c = ((1, 3), (2, 5), (4, 7), (6, 9), (8, 10)). It follows then that $\mathbf{F}_2 = \{(f_a, f_b), (f_a, f_c)\}$, where $f_a < f_b < f_c$.

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

(1) $f_1f_2 = f_af_b$ (type '46'):

We map any f_jf_k, $1 \le j < k \le i+1$ of type '46' into f_af_b (in such a way that either f_j or f_k is mapped to f_a). To map into *any other* two one-factors of type '46' would always make $F_i^{\alpha} > F_i$ (and hence would not tell us whether F_i is not canonical). There are 2·(2·2)·(2·3) = 48 ways to do this.

We may ignore those f_jf_k of type '10', as mapping them into f_af_c would always make $F_i^{\alpha} > F_i$. (In general , if f_1f_2 is of type 'x', we may ignore f_jf_k of type 'y' so long as the canonical pair of one-factors corresponding to type 'y' are greater than those of type 'x'. See the following section.) The maximum number of mappings α required in this case is 48·(9·8)/2 =1728, which is 1/20 as many mappings used when mapping a one-factor to another (which needs 34560 mappings).

(2) $f_1f_2 = f_af_c$ (type '10'):

All f_jf_k, $1 \le j < k \le i+1$, must be of type '10' (in general, no f_jf_k can be of a type corresponding to a canonical structure less than f_1f_2). Thus we discard those $f \in \mathbf{U}_{i+1}$ which form a type '46' structure with any of $f_j \in F_i$, $1 \le j \le i$, 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 $20{\cdot}(9{\cdot}8)/2 = 720$.

Table 5.1 gives the number of canonical proper partial OFs and CPU time taken for each of the steps. The number of (complete) OFs of K_{10} agrees with the results in Gelling [22]. 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. The enumeration took approximately 10.5 minutes of CPU time.

5.3 One-factorizations of K_{12}

We could use the same algorithms in the previous section to construct the canonical (non-isomorphic) OFs of K_{12}. However, the number of canonical structures grows at such an astronomical rate that it is infeasible to have a complete enumeration at this point in time. This is illustrated in Table 5.2, where we use breadth-first algorithm to enumerate sets of canonical proper partial OFs $\mathbf{F}_i$ (i = 2, 3, and 4) of K_{12} containing a sub-OF of K_4.

Table 5.1

Non-isomorphic canonical proper partial OF of K_{10}

Step i+1	type '46'	type '10'	total	CPU time (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

Table 5.2

Non-isomorphic canonical proper partial OF of K_{12} containing sub-OF of K_4

Step i+1	# of canonical structures	CPU time (minutes)
3	6	0.1
4	295	0.7
5	15445	26.0

In the remainder of this section, we describe how the different structures formed by a pair of distinct one-factors of K_{2n} may be incorporated in the orderly algorithms in Chapter 2 for enumerating OFs of K_{2n}. We use K_{12} as an example.

First of all, we determine the different types of cycle structure that can exist for a pair of distinct one-factors of K_{2n}. We note that the number of different structures of a pair of distinct one-factors of K_{2n} increases with n. Thus, a pair of distinct one-factors of K_{12} forms either (i) 3 cycles of length 4, or (ii) 1 cycle of length 4 and 1 cycle of length 8, or (iii) 2 cycles of length 6, or (iv) a Hamiltonian cycle.

We then find the smallest one-factors in $\mathbf{U}_2$ that form such cycle structures with f_a, the smallest one-factor of K_{2n}. For K_{12}, the four one-factors are:

(i) f_i = ((1, 3), (2, 4), (5, 7), (6, 8), (9, 11), (10, 12));

(ii) f_{ii} = ((1, 3), (2, 4), (5, 7), (6, 9), (8, 11), (10, 12));

(iii) f_{iii} = ((1, 3), (2, 5), (4, 6), (7, 9), (8, 11), (10, 12));

(iv) f_{iv} = ((1, 3), (2, 5), (4, 7), (6, 9), (8, 11), (10, 12)).

Note that $f_a < f_i < f_{ii} < f_{iii} < f_{iv}$.

We say that a pair of distinct one-factors is of type 'x' if it is isomorphic to the one-factors f_a and f_x. We can now define an ordering on the types of cycle structure for a pair of distinct one-factors. We say a pair of distinct one-factors of type 'x' < a pair of distinct one-factors of type 'y' if $f_x < f_y$, where f_x, f_y are the smallest one-factors in $\mathbf{U}_2$ that forms type 'x' and type 'y' cycle structures with f_a respectively. The breadth-first algorithm in Section 2.5 can then be modified as follows (modifications to the depth-first algorithm are similar):

(1) In considering whether $F_i = \{f_1, f_2, ..., f_i\}$ could be extended to $F_i \cup \{f\}$, if for some $1 \leq j \leq i$, the type of $\{f_j, f\}$ < type of $\{f_1, f_2\}$, then $F_i \cup \{f\}$ is not

canonical and we do not extend F_i to $F_i \cup \{f\}$. Otherwise, go to (2).

(2) (i) For $1 \le j \le i$, if the type of $\{f_j, f\}$ is greater than the type of $\{f_1, f_2\}$, then we do not have to carry out the mapping α of $\{f_j, f\}$ into $\{f_1, f_2\}$, since $(F_i \cup \{f\})^\alpha > F_i \cup \{f\}$.

(ii) For $1 \le j \le i$, if the type of $\{f_j, f\}$ is equal to the type of $\{f_1, f_2\}$, we perform the mapping α of $\{f_j, f\}$ into $\{f_1, f_2\}$. If $(F_i \cup \{f\})^\alpha < F_i \cup \{f\}$ for some j, $F_i \cup \{f\}$ is not canonical and we do not extend F_i to $F_i \cup \{f\}$.

5.4 Cycle structures of automorphism groups of one-factorizations of K_{2n}

As a complete enumeration of non-isomorphic OFs of K_{12} is not possible at this point in time, we turned to a problem of a smaller scale: non-isomorphic OFs containing prescribed (non-trivial) automorphism groups.

For the remainder of this chapter, we describe how automorphism orderly algorithms as described in Chapter 3 are used to enumerate OFs of K_{12} with certain prescribed automorphism groups.

Let a be a permutation of {1, ..., 12}, and define $A = \langle a \rangle$. The generator a of a cyclic group A on 12 elements can have one of 77 different cycle structures (see Appendix 1). Many of these cases can be eliminated easily by the following general results on the cycle structure of automorphisms of OFs of K_{2n}.

Lemma 5.2 If a is a non-identity automorphism of an OF of K_{2n}, then the number of fixed points in a is even or 1.

Proof Let the number of fixed points of a be 2k+1 ($k \ge 1$), and let the fixed points be $p_1, p_2, p_3, \ldots, p_{2k+1}$. Consider the one-factor f containing the edge $\{p_1, p_2\}$. Then f must be an orbit of length one. But then there exists an edge $\{p_i, p_j\}$ (q_j is a non-fixed point) in f which maps into an edge of another one-factor; hence a contradiction.

Lemma 5.3 If *a* is a non-identity automorphism of K_{2n} and has more than n fixed points, then the number of fixed points in *a* = 2n.

Proof Let the number of fixed points be 2k, where 2k > n. Then there exists an edge of two fixed points in every one-factor of the OF. Every one-factor is thus an orbit of length one. Consequently, each one-factor has k edges made up of the 2k fixed points and it is impossible to have an edge of the form $\{p_i, q_j\}$, where p_i is a fixed point and q_j is a non-fixed point (except the case when all the points in *a* are fixed points).

Lemma 5.4 If *a* is a non-identity automorphism of an OF of K_{2n} and has exactly n fixed points, then the remaining n points of *a* must appear as disjoint 2-cycles.

Proof Consider the one-factors that are fixed by *a*. Each of these one-factors has n/2 edges made up of fixed points, so there are exactly n-1 such one-factors.

The remaining n one-factors consist of edges of the form $\{p_i, q_j\}$, where p_i is a fixed point and q_j is a non-fixed point. Therefore, all edges made up of non-fixed points, $\{q_i, q_j\}$, must appear in the n-1 fixed one-factors. Consequently, the non-fixed points can only appear as disjoint 2-cycles in *a*.

Corollary If $n \equiv 2 \pmod 4$, then the number of fixed points in *a* cannot be n (except when $n = 2$).

Proof Consider the n-1 one-factors that are fixed by *a*. Each of these one-factors has n/2 edges from the n points in the 2-cycles. Since $n \equiv 2 \pmod 4$, each of these one-factors must have at least one edge of the form $\{p_1, p_2\}$, where p_1 and p_2 appear in the same 2-cycle. Now there are n/2 2-cycles (edges) to be filled in these n-1 one-factors. So $(n/2) \geq (n-1)$, or $n \leq 2$.

Lemma 5.5 If *a* is a non-identity automorphism of an OF of K_{2n} and has no fixed points, then the number of 3-cycles in *a* cannot be 1.

Proof Consider a 3-cycle (a b c). Edges {a, b}, {b, c} and {c, a} appear in 3 distinct one-factors forming an orbit of length 3. Thus we have

{a, b} --> {b, c} --> {c, a} --> {a, b}

{c, x} --> {a, y} --> {b, z} --> {c, x} ;

and (x y z) is another 3-cycle.

Lemma 5.6 Let *a* be a non-identity automorphism of an OF of K_{2n}. If the number of fixed points in *a* is even and the remaining points form a cycle, then there must be exactly two fixed points in *a*.

Proof Let the number of fixed points be 2k, then the number of non-fixed points is 2n-2k and they form a cycle $(q_1\ q_2\ q_3 \ldots q_{2n-2k})$.

Consider the one-factors containing the edge $\{p_i, p_j\}$ made up of fixed points: there are 2k-1 of these one-factors (orbits). There is only one way that the non-fixed points may appear in these 2k-1 one-factors; they must appear as edges $\{q_1, q_{n-k+1}\}$, $\{q_2, q_{n-k+2}\}$, $\{q_3, q_{n-k+3}\}$, ..., and $\{q_{n-k}, q_{2n-2k}\}$. Thus 2k-1 = 1 and hence the number of fixed points is two.

Lemma 5.7 Let *a* be a non-identity automorphism of an OF of K_{2n}. If there is a 2-cycle (a b) in *a*, then the OF has an orbit of length one.

Proof The one-factor containing the edge {a, b} must be fixed by *a*.

Corollary If *a* has exactly one fixed point, then there cannot be any 2-cycles in *a*.

Proof If *a* has exactly one fixed point, then there is no one-factor fixed by *a*. Consequently, there cannot be any 2-cycles in *a*.

Lemma 5.8 Let *a* be a non-identity automorphism of an OF of K_{2n}. If *a* has 2 cycles of lengths L1 and L2 (L1 < L2), then LCM(L1, L2) $\leq$ 2n-1.

Proof Let the L1-cycle be denoted $(p_1\ p_2 \ldots p_{L1})$, and the L2-cycle be denoted by $(q_1\ q_2 \ldots q_{L2})$. Since L1 $\neq$ L2, the one-factor f containing the edge $\{p_1, q_1\}$ is in an orbit of length greater than one. So f maps into another one-factor containing the edge $\{p_2, q_2\}$, which in turn maps into the one-factor containing the edge $\{p_3, q_3\}$, and so on. Thus the one-factor containing $\{p_1, q_1\}$ is in an orbit of length LCM(L1, L2). Hence, LCM(L1, L2) $\leq$ 2n-1.

5.5 One-factorizations of K_{12} containing prescribed automorphism groups

We have the following theorem on the cycle structures that admit OFs for K_{12}.

Theorem 5.9 There are at most 18 cycle structures of *a* that admit OFs for K_{12}.

Proof Using Lemmas 5.2 to 5.8, we eliminated all but 29 cases (refer to Appendix 1). Of these 29 cases, we can eliminate 11 further cases, by observing that a^n for some n >1 is not an admissible automorphism. As an example, for case 24, *a* has cycle structure $6^1 3^2$. But then a^3 has cycle structure $2^3 1^6$, which is case 74 and is ruled out by Lemma 5.4.

For those cases that are not eliminated by the above lemmas and hence may admit OFs, we resort to the help of computer. All the cases in Appendix 1 except cases 71 and 77 are dealt with. For cases 72 and 73, we first used the breadth-first algorithm to construct $\mathbf{F}_4$, then extended the proper partial OFs in $\mathbf{F}_4$ to complete OFs by the depth-first algorithm. For the other cases, only the depth-first algorithm was used.

Both cases 71 and 77 require large amount of computing time. Case 77 is equivalent to a complete enumeration of the OFs of K_{12}, which is out of our reach at this point in time. Case 71 involves constructing OFs containing automorphisms of six 2-cycles. Instead of dealing with case 71 in its entirety, we looked at the subproblem of the enumeration of OFs that contain two automorphisms of six 2-cycles. That is, we have A = < a_1, a_2 >, where a_1 = ((1 2) (3 4) (5 6) (7 8) (9 10) (11 12)) and a_2 = ((1 3) (2 4) (5 7) (6 8) (9 11) (10 12)). (It turns out that, up to isomorphism, this is the only admissible case once we pick a_1.) We refer to this as case 78 in Table 5.4. Similar to cases 72 and 73, we used the combination of breadth-first and depth-first algorithms.

Therefore, we have enumerated all OFs of K_{12} except those containing exactly one automorphism of six 2-cycles and those with the trivial automorphism group.

In Table 5.4, we list the cases that admit at least one OF and the associated statistics. It is interesting to note that there are 6 cases where N(A) did not eliminate all the isomorphic OFs (cases 20, 44, 46, 47, 73 and 78).

Interested in finding out what mappings would have eliminated these isomorphic OFs, we looked into case 20, where 30 pairs of isomorphic OFs survived the test of N(A). Here, A = < (1 2 3 4 5 6) (7 8 9 10 11 12) >.

Of these 30 pairs of OFs, 6 of them have the *full* automorphism groups of order 12, and 21 have order 24. The automorphism groups of these 27 pairs of OFs each contains a unique cyclic subgroup B = < (1 3 5) (2 4 6) (7 9 11) (8 10 12) >. Since B is unique, any α that takes an OF into its isomorphic copy must also maps B into B; that is, $\alpha \in$ N(B). Thus if we use N(B) instead of N(A), we would be able to eliminate the 27 isomorphic OFs.

Each of the remaining 3 pairs of OFs has the full automorphism group of order 48, and each has 4 copies of $\mathbf{Z}_3$ in its automorphism group. In each of these three cases, there exists an $\alpha \in N(B)$ which takes a OF into its isomorphic copy. Here again, using N(B) would have eliminated the 3 isomorphic OFs.

It should be emphasized that, in general, we do not know what the full automorphism groups look like beforehand. Consequently, the best strategy is perhaps to use the normalizer of the prescribed subgroup N(A) to obtain the quasi-A-canonical OFs, followed by testing these OFs for isomorphism. The statistics on K_{12} indicates that N(A) is able to get rid of most of the isomorphic OFs. We would like to point out that, in certain situations, however, the normalizer of the prescribed group N(A) is sufficient to eliminate isomorphic OFs; that is, quasi-A-canonical OFs are non-isomorphic in these cases. These results were derived in [33] and [46] which we record as the following theorem:

Theorem 5.10 Suppose two OFs F and G of K_{n+1} contain $\mathbf{Z}_n$ in their automorphism groups, where n is an odd prime or the product of two distinct primes. If F is isomorphic to G, then $F^{\alpha} = G$, for some $\alpha \in N(\mathbf{Z}_n)$.

Thus for $A = \mathbf{Z}_{11}$, the OFs of K_{12} constructed with the use of N(A) are non-isomorphic (case 2 in Table 5.4 and Appendix 1).

Table 5.3 gives the distribution of the orders of automorphism groups for the OFs of K_{12} constructed in this paper. Note that the numbers in Table 5.3 are exact, with the exception of the number of OFs of order 2.

The CPU time for all cases dealt with except 72, 73 and 78 added up to about 40 minutes. Case 72 took 7.5 hours, case 73 needed 30 hours, and case 78 consumed about 17 hours. These timings include the final step to eliminate the isomorphic OFs from the set of quasi-A-canonical OFs (we first find the *canonical* representation of these OFs, and then delete any duplications).

Given enough computer time, it would appear that case 71 can be completely resolved. It remains to be seen how long it will take to enumerate the case of trivial automorphism (case 77). However, judging from the fact that there are 298 non-isomorphic automorphism-free OFs of K_{10} (out of a total of 396) [22], we suspect there will be many more non-isomorphic automorphism-free OFs for K_{12}. (It is interesting to note that none of the complete graphs of lower order has automorphism-free OFs.) In fact, it has been shown in [5] and [41] that the number of non-isomorphic automorphism-free OFs of K_{2n} increases rapidly and goes to infinity with n. (In [41], it is also shown that an automorphism-free OF of K_{2n} exists if and only if $n \geq 5$.)

Table 5.3

Frequency distribution of the orders of automorphism groups of OFs of K_{12}

Order	No.
2	≥ 39706
3	669
4	14801
5	92
6	245
8	610
10	10
11	2
12	138
16	76
20	2
24	25
32	4
48	6
55	1
110	1
240	2
660	1
Total	≥56391

Table 5.4

One-factorizations of K_{12} containing prescribed automorphism groups

Case No.	Cycle Structure of *a*	$\|N(A)\|$	No. of distinct orbits	No. of distinct 1-factors	Quasi-A-canonical OF Total (1)	Quasi-A-canonical OF Not in prev. cases (2)	Quasi-A-canonical OF In prev. cases (3)	A-canonical OF Not in prev. cases (4)	A-canonical OF In prev. cases (5)
1	12^1	48	19	79	6	6	0	6	0
2	$11^1 1^1$	110	25	275	5	5	0	5	0
3	$10^1 2^1$	80	17	81	3	2	1	2	1
4	$10^1 1^2$	80	57	561	7	6	1	6	1
11	$8^1 2^1 1^2$	128	133	1033	12	12	0	12	0
20	6^2	144	221	1073	297	287	10	227	8
32	$5^2 1^2$	400	905	4505	109	97	12	97	12
44	4^3	768	709	2557	390	381	9	376	8
46	$4^2 2^2$	512	399	1551	76	74	2	64	2
47	$4^2 2^1 1^2$	256	565	2213	328	291	37	273	31
48	$4^2 1^4$	1536	783	3087	222	173	49	173	49
59	3^4	3888	1953	5805	1086	850	236	850	236
72	$2^5 1^2$	7680	2561	5041	5676	5665	11	5665	11
73	$2^4 1^4$	9216	1803	3531	38751	38029	722	37063	598
78	$2^6 x 2^6$	2304	399	927	13341	11572	1769	11572	695
Total								56391	

(1) = (2) + (3).

(2) - (4) gives the number of isomorphic OFs (not appearing in previous cases) which are not eliminated by N(A).

(3) - (5) gives the number of isomorphic OFs (appearing in previous cases) which are not eliminated by N(A).

CHAPTER 6

ENUMERATING PERFECT ONE-FACTORIZATIONS OF K_{14}

6.1 Introduction

In this chapter, we investigate the number of non-isomorphic perfect OFs of K_{14}. Four non-isomorphic perfect OFs for K_{14} were shown to exist in [43]. Using the orderly algorithms described in Chapters 2 and 3, we construct 17 new perfect OFs and hence improve the lower bound to 21. We also show that these are the only perfect OFs of K_{14} having non-trivial automorphism groups.

Theorem 6.1 $N_p(14) \geq 21$.

We also compute the automorphism groups of these perfect OFs. We find examples where the automorphism group has order 2, 3, 4, 6, 12, 84, and 156. It is interesting to note that none of the OFs is automorphism-free. The existence of an automorphism-free perfect OF for K_{12} would lead one to suspect there might be some of these for K_{14}.

6.2 Orderly algorithms

In this and the following sections, we describe how the orderly algorithms of Chapter 2 are used to construct perfect OFs of K_{12} and K_{14}. In the later sections, we apply automorphism orderly algorithms of Chapter 3 to produce perfect OFs of K_{14} containing certain automorphism groups.

In testing whether a proper partial perfect OF $F_i \cup \{f\}$ of K_{2n} is canonical, we used those α's that map a pair of distinct one-factors of $F_i \cup \{f\}$ into a fixed pair of one-factors. Since we require any pair of disjoint one-factors to form a Hamiltonian cycle, we can map a Hamiltonian cycle of length 2n into another one of the same length. The number of such mappings is $2 \cdot 2n = 4n$. In the case of K_{14}, we map any pair of disjoint one-factors into f_a = ((1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14)) and f_b = ((1, 3), (2, 5), (4, 7), (6, 9), (8, 11), (10, 13), (12, 14)), where f_b is the smallest one-factor in $\mathbf{U}_2$ that forms a Hamiltonian cycle with f_a. The maximum number of such mappings for K_{2n} is $(2n-1) \cdot 4n$. Thus we have a polynomial-time algorithm for determining isomorphism of perfect OFs of complete graphs. In general, it is unknown if one can determine isomorphism of OFs in polynomial time. The best known algorithms have complexity $n^{c(\log n)}$ (see [16]).

6.3 Results on K_{12} and K_{14}

We started out by implementing the breadth-first algorithm, since this method tells us the number of non-isomorphic proper partial perfect OFs at each intermediate level before proceeding to the next level. It took approximately 132 minutes of CPU time to construct the 5 perfect OFs of K_{12}. Using the depth-first method, incorporating pruning as described in Sections 2.9 and 2.10, the number of intermediate proper partial perfect OFs is significantly reduced , and the enumeration took only 23 minutes of CPU time.

(Depth-first algorithm without pruning took approximately 50 minutes of CPU time.) The following table gives the number of canonical proper partial perfect OFs of each rank, both with and without pruning.

Table 6.1

Non-isomorphic canonical proper partial perfect OFs of K_{12}

i	# of canonical proper partial perfect OFs of rank i	
	without pruning	with pruning
3	24	24
4	395	395
5	2679	2679
6	10987	10906
7	13791	3542
8	3491	14
9	209	7
10	6	6
11	5	5

It is interesting to note that a canonical partial perfect OF need not be proper. For example, there are exactly 32 (non-isomorphic) canonical partial perfect OFs of rank 3 (see [34]), but only 24 of these are proper (see Table 6.1).

When we used the breadth-first method to attempt to enumerate $N_p(14)$, it did not take long for us to conclude that the complete enumeration is impossible at this time. The number of proper partial perfect OF structures generated and the amount of CPU time increase dramatically from one step to the next, as indicated by the following table.

Table 6.2

Non-isomorphic canonical proper partial perfect OFs of K_{14} using the breadth-first algorithm

i	# of canonical proper partial perfect OFs of rank i
3	174
4	23704
5*	34272

* using only the first 464 sets $F_4 \in \mathbf{F}_4$

Consequently, we decided to improve the lower bound on $N_p(14)$ by constructing as many perfect OFs of K_{14} as possible. We used the breadth-first method to construct all partial perfect OFs in $\mathbf{F}_4$. Then, given a partial perfect OF in $\mathbf{F}_4$, the depth-first method was used to generate all extensions to complete perfect OFs.

With this approach, we were able to find 11 new perfect OFs of K_{14}. They are listed as sets 2 - 12 in Appendix 2. The four previously known perfect OFs of K_{14} are sets 1, 13, 14, and 15 (see Appendix 2). Sets 1 and 13 are GA_{14} and GK_{14} respectively. Sets 13, 14 and 15 are constructed from even-starters in $\mathbf{Z}_{12}$; and set 13 can also be generated by a starter in $\mathbf{Z}_{13}$.

In total, 105 hours of CPU time were spent in finding these 11 new perfect OFs.

6.4 Perfect one-factorizations of K_{14} containing prescribed automorphism groups

The algorithms outlined in Chapter 3 were also to construct perfect OFs of K_{14} that contain prescribed automorphism groups.

We were able to prove that there are exactly 21 perfect OFs of K_{14} with non-trivial automorphism groups. In fact, these algorithms helped find 6 new perfect OFs (sets 16 - 21 in Appendix 2), in addition to the 11 new perfect OFs found by the orderly algorithms of Chapter 2.

We started by looking at the cycle structures of a permutation *a* of 14 elements.

In total, there are 135 possibilities, many of which can be easily eliminated. In fact, Ihrig proved the following results in [29].

Lemma 6.2 ([29], Theorem 3.3)
If *a* is a non-identity automorphism of a perfect OF of K_{2n}, then the number of fixed points is at most 2.

Proof Suppose the number of fixed points of *a* is 2s, where $s \geq 1$ (from Lemma 5.2 we know that it must be even). There are exactly s(2s-1) edges made up of these fixed points, and these edges appear only in one-factors which are fixed by *a*. Thus there are precisely s(2s-1)/s = 2s-1 one-factors fixed by

a, since each such one-factor contains s edges from the 2s fixed points. Since s > 1, there exist at least one such one-factors. The edges of any two of these one-factors containing the 2s fixed points form a cycle of length at most 2s. But then the OF is not perfect (unless 2s = 2n).

Lemma 6.3 ([29], Corollary 3.4)

If a is an automorphism of a perfect OF of K_{2n}, then its cycle structure must be one of the 4 forms:

(1) $1^2k^{(2n-2)/k}$, (2) $1^1k^{(2n-1)/k}$, (3) $2^1k^{(2n-2)/k}$, (4) $k^{2n/k}$.

Theorem 6.4 There are at most 13 cycle structures of a that admit perfect OFs for K_{14}.

Proof By Lemma 6.3, there are at most 14 cycle structures of a that admit perfect OFs for K_{14}. They are:

(1) 14^1	(2) 13^11^1	(3) 12^12^1	(4) 12^11^2
(5) 7^2	(6) 6^22^1	(7) 6^21^2	(8) 4^32^1
(9) 4^31^2	(10) 3^42^1	(11) 3^41^2	(12) 2^7
(13) 2^61^2	(14) 1^{14}.		

Case 10 can be eliminated since a^3 has the form 2^11^{12}, which is not admissible.

We list in Table 6.3 those cases that admit at least one perfect OF for K_{14} and the associated statistics. We omit the case involving the trivial automorphism (case 14), as this would amount to a complete enumeration. In total, approximately 10 hours of computer time was required for the remaining 12 cases (including the time required to determine the canonical representation of the perfect OFs constructed).

Table 6.3

Perfect OFs of K_{14} containing prescribed automorphism groups

Case No.	Cycle Structure of *a*	|N(A)|	No. of distinct orbits	No. of distinct 1-factors	- Non-isomorphic perfect OF - total generated	set no. in Appendix 2
1	14^1	84	12	63	1	set 1
2	$13^1 1^1$	156	1	13	1	set 13
4	$12^1 1^2$	96	25	289	3	sets 13, 14, 15
5	7^2	294	565	3913	1	set 1
7	$6^2 1^2$	288	1399	8359	12	sets 1, 3 - 6, 9 - 15
9	$4^3 1^2$	1536	4621	18445	5	sets 9, 10, 13 - 15
11	$3^4 1^2$	7776	15579	46683	17	sets 1, 3 - 6, 9 - 20
12	2^7	645120	23880	46920	4	sets 1, 2, 3, 21
13	$2^6 1^2$	92160	32395	64659	15	sets 1 - 15

Of special interest is case 12 (where $a = ((1\ 2)\ (3\ 4)\ \ldots\ (13\ 14))$). It is not difficult to see that the seven edges from the seven 2-cycles of *a* must either (i) appear in the same one-factor, or (ii) appear in seven distinct one-factors. An

OF of type (i) would have 1 orbit of length 1 and 6 orbits of length 2, and type (ii) would have 7 orbits of length 1 and 3 orbits of length 2. In using orderly algorithms for these two subcases, we omit canonicity testing (645120 mappings would have been needed for each F_i), and test the OFs for isomorphism after they have been created.

In [29], Ihrig defined a P element to be an automorphism of n 2-cycles of a perfect OF of K_{2n}, the cycles of which form the edges of a one-factor of the perfect OF. Thus, an OF of type (i) contains a P element. Ihrig observed that, other than the perfect OF of K_4, there is no other example known of a perfect OF of K_{2n} containing a P element. Our computer search did not find any perfect OF of type (i) for K_{14}. In Section 6.6, we will show that, when n is even (except n = 2), there does not exist a perfect OF of K_{2n} containing a P element; hence the smallest unknown case is K_{18}.

There are 165 perfect OFs of type (ii), of which 4 are non-isomorphic. The information on the number of orbits and distinct one-factors listed in Table 6.3 for case 12 pertains to type (ii).

It is interesting to note that, except for case 12, the quasi-A-canonical perfect OFs constructed from each of the other cases turn out to be non-isomorphic (that is, they are also A-canonical).

6.5 Automorphism groups of perfect one-factorizations of K_{14}

We summarize from Appendix 2 the automorphism groups of the 21 perfect OFs of K_{14}:

$\mathbb{Z}_2$ is the automorphism group for sets 7, 8 and 21;

$\mathbb{Z}_3$ is the automorphism group for sets 16 - 20;

$\mathbb{Z}_2 \times \mathbb{Z}_2$ is the automorphism group for set 2;

$\mathbb{Z}_6$ is the automorphism group for sets 4 - 6, 11, and 12;

$\mathbb{Z}_2 \times \mathbb{Z}_6$ is the automorphism group for set 3;

$\mathbb{Q}_6$ (a dicyclic group) is the automorphism group for sets 9 and 10;

$\mathbb{Z}_{12}$ is the automorphism group for sets 14 and 15;

$[\mathbb{Z}_{13}]\, \mathbb{Z}_{12}$ (semi-direct product) is the automorphism group for set 13; and

$[\mathbb{Z}_{14}]\, \mathbb{Z}_6$ (semi-direct product) is the automorphism group for set 1.

In [29], Ihrig studies automorphism groups of perfect OFs. The next two theorems give several properties such a group must have, if it contains an automorphism of order 2 having fixed points.

Theorem 6.5 ([29], Theorem 5.5)

If a perfect OF for K_{2n} contains a noncentral automorphism of order 2 having fixed points, then the perfect OF is either GK_{2n} (2n-1 prime) or GA_{2n} (n prime).

Theorem 6.6 ([29], Theorem 5.9)

If a perfect OF on K_{2n} contains a central automorphism of order 2 having fixed points, then the following statements hold:

(a) the order of the automorphism group divides 2n-2;

(b) there are at most 3 automorphism of order 2, and only one of these has fixed points.

We note that the perfect OFs of sets 1 - 15 and 21 all have an automorphism of order 2 containing fixed points. Our examples illustrate every group order allowed by Theorem 6.6 (a) (namely, orders 2, 4, 6, and 12). Also, note that sets 2 and 3 each contain three automorphisms of order 2, while sets 4 - 12 and 21 each contain only one such automorphism.

If the automorphism group of a perfect OF does not contain an automorphism of order 2 containing fixed points, then the following results hold.

Theorem 6.7 ([30], Theorem 3.10)

If a perfect OF on K_{2n} contains no automorphism of order 2 having fixed points, then the order of the automorphism group is $m_0 \cdot m_1 \cdot m_2$, where $m_0 \mid 2n$, $m_1 \mid (2n - 1)$, and $m_2 \mid (n - 1)$. Further, m_2 is odd; and at least one of m_0, m_1, and m_2 is equal to 1.

In the case of K_{14} we obtain $m_0 \mid 14$, $m_1 \mid 13$, and $m_2 \mid 3$. If $m_1 = 13$, then the perfect OF must be generated by a starter in $\mathbf{Z}_{13}$. These were enumerated in [2], and set 13 (GK_{14}) is the only example. Hence, we can ignore this case, and assume $m_1 = 1$. Then, the order of the automorphism group must divide 42.

We have enumerated all perfect OFs of K_{14} having an automorphism of order 7, and set 1 is the only example. Consequently, the order of the automorphism group must divide 6, and orders 1 and 3 are the only new possibilities. As mentioned already, sets 16 - 20 have automorphism groups isomorphic to $\mathbb{Z}_3$, and we have no examples with trivial automorphism groups.

Hence, we have examples of every possible group order, except 1.

Two of the new perfect OFs found (sets 9 and 10) have the property that their automorphism groups are the dicyclic group Q_6 of order 12. The dicyclic group Q_{2n} of order 4n is the group defined by $Q_{2n} = \{ a^i b^j : 0 \leq i \leq 2n-1, 0 \leq j \leq 1, a^{2n} = e, b^2 = a^n, bab^{-1} = a^{-1} \}$.

A *sequencing* of a finite group G of order 2n is an ordering e, $a_2, a_3, \ldots, a_{2n}$ of all elements of G such that the partial products e, $ea_2, ea_2a_3, \ldots, ea_2 \cdots a_{2n}$ are distinct and hence also all of G. The sequencing is *symmetric* if in addition the following are true: (1) G has a unique element z of order 2, (2) $a_{n+1} = z$, and (3) $a_{n+1+i} = (a_{n+1-i})^{-1}$. In [4], Anderson observed that these two perfect OFs (sets 9 and 10) give rise to symmetric sequencings in the group Q_6. (A sequencing in this group was previously unknown). Subsequently, Anderson[4] showed that for any odd $n \geq 3$, the dicyclic group Q_{2n} can be symmetrically sequenced.

Given a symmetric sequencing of a group G, one can construct an OF (not necessarily perfect) of $K_{|G|+2}$ (see [4]). It thus seems hopeful that symmetric sequencings of Q_{2n} can be used to construct perfect OFs of K_{4n+2}. However, it remains to be seen whether symmetric sequencings will give us a new class of perfect OFs.

6.6 Perfect one-factorizations of K_{2n} containing a P element

In this section, we investigate the perfect OFs of K_{2n} containing a P element (refer to Section 6.4 for definition), and prove that such perfect OFs do not exist for K_{2n} when n is even (with the exception of n = 2). For n odd, it remains an open problem, and the smallest unknown case is 2n = 18.

Without loss of generality, assume the P element of K_{2n} is a = (1 2) (3 4) ... (2n-1 2n). Hence, a perfect OF of K_{2n} containing the P element has f = ((1, 2), (3, 4), ..., (2n-1, 2n)) as a one-factor, and has the orbit structure as described in the following lemma.

Lemma 6.8 If n > 2, then a perfect OF of K_{2n} containing a P element has 1 orbit of length 1 and (n-1) orbits of length 2, under the action of a.

Proof f is an orbit of length 1.

Assume there exists another one-factor g which is an orbit of length 1. Without loss of generality, suppose g contains the edge {1, 3}. Now it must map into {2, 4} in g. But then we have a 4-cycle on vertices {1, 2, 3, 4}, and the OF is not perfect (unless $n \leq 2$).

Let O = $\{f_1, f_2\}$ be an orbit of length 2 under the action of a. By definition, the 2n edges of O form a Hamiltonian cycle. Without loss of generality, we can name the vertices on the cycle (in the clockwise direction) by $a_0, a_1, a_2, ..., a_{2n-1}$. (Thus $\{a_i : i = 0, 1, ..., 2n-1\} = \{i : i = 1, 2, ..., 2n\}$.) The edges of f (the orbit of length 1) must then be of the form given in the following lemma.

Lemma 6.9 The one-factor $f = \{\{a_i, a_{n+i}\} : i = 0, 1, ..., n-1\}$.

Proof Suppose $\{a_0, a_k\} \in f$. Consider an edge $\{a_0, a_1\} \in f_1$: it must map into an edge $\in f_2$. It maps into either (i) $\{a_k, a_{k+1}\}$, or (ii) $\{a_k, a_{k-1}\}$.

In case (i), $\{a_1, a_{k+1}\} \in f$. But then $\{a_1, a_2\}$ maps into $\{a_{k+1}, a_{k+2}\}$, and thus $\{a_2, a_{k+2}\} \in f$. Using similar arguments, we can show that $\{a_3, a_{k+3}\}$, $\{a_4, a_{k+4}\}$, ..., $\{a_k, a_{k+k}\} \in f$. Now $\{a_0, a_k\} = \{a_k, a_{k+k}\}$, and hence $2k = 0$, or $k = n$. Thus $f = \{\{a_i, a_{n+i}\} : i = 0, 1, ..., n-1\}$.

In case (ii), $\{a_1, a_{k-1}\} \in f$. But then $\{a_1, a_2\}$ maps into $\{a_{k-1}, a_{k-2}\}$, and thus $\{a_2, a_{k-2}\} \in f$. Using similar arguments, we can show that $\{a_3, a_{k-3}\}$, $\{a_4, a_{k-4}\}$, ... $\in f$. If k is even, this implies $\{a_{k/2}, a_{k/2}\} \in f$ and *a* has a fixed point, which is impossible (by definition). If k is odd, we have $\{a_{(k-1)/2}, a_{(k+1)/2}\} \in f$. But then $\{a_{(k-1)/2}, a_{(k+1)/2}\}$ is also an edge of O; a contradiction.

We now prove the main result of this section.

Theorem 6.10 For n even and > 2, there does not exist a perfect OF of K_{2n} containing a P element.

Proof Let $n = 2m$. From Lemma 6.9, we know that the edge $\{a_0, a_1\}$ maps into the edge $\{a_{2m}, a_{2m+1}\}$. It is easy to see that edges $\{a_0, a_1\}$ and $\{a_{2m}, a_{2m+1}\}$ appear in the same one-factor. Now, $\{a_0, a_{2m}\}$ and $\{a_1, a_{2m+1}\}$ are two edges of f. Hence, there exists a 4-cycle on $\{a_0, a_1, a_{2m}, a_{2m+1}\}$, and the OF is not perfect.

CHAPTER 7

ENUMERATING ONE-FACTORIZATIONS AND HOWELL DESIGNS OF K_{10} MINUS A ONE-FACTOR

7.1 Introduction

Using the orderly algorithms in Chapters 2 and 4, we enumerate the (non-isomorphic) OFs and sets of orthogonal OFs of the graph K_{10} - f, where f is a one-factor of K_{10}. We find that there are 3192 OFs; 18220 pairs, 3 triples, and 1 quadruple of mutually orthogonal OFs. It is also shown that there is no set of five mutually orthogonal OFs.

7.2 A conjecture about the number of orthogonal OFs of regular graphs

The results about K_{10} - f are interesting for several reasons. First, the non-isomorphic OFs and Howell designs have been enumerated for all graphs on at most 10 vertices except K_{10} - f (see [21], [22] and [51]). Hence, the results of this chapter complete this census. Also, the graph K_{10} - f is the smallest graph (other than complete or complete bipartite graphs) for which there exist three (or more) orthogonal OFs.

It has been conjectured that the maximum number of mutually orthogonal OFs of a regular graph on n vertices is at most (n-2) / 2. (It has been shown in [24] that the maximum number of orthogonal OFs of K_{2n} goes to infinity with n.)

There are in fact infinitely many graphs for which (at least) (n-2) / 2 mutually orthogonal OFs are known to exist, but there are no graphs known for which this conjectured bound is exceeded. The following results were previously known.

Theorem 7.1 The following graphs have *at least* (n-2) / 2 orthogonal OFs:

(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 [21] and [24]. The OFs of the graphs in (2) are equivalent to mutually orthogonal Latin squares, so this result is well-known. The result (3) is proved in [24].

The four orthogonal OFs 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 OFs. 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.

7.3 Orderly algorithms and canonicity mappings

The results are established with the use of the orderly algorithms described in Chapters 2 and 4. In particular, we used the breadth-first algorithm.

Without loss of generality, we let $f = f_a = ((1, 2), (3, 4), ..., (11, 12))$, the smallest one-factor of K_{10}. We note that the OFs of K_{10} - f have eight one-factors and do not include the five edges in f.

In constructing the OFs, we pretend that f_a is part of the OFs of K_{10} - f. That is, the set of proper partial OF of rank 1, $\mathbf{F}_1$, is $\{f_a\}$. The set of proper partial OF of various ranks are then constructed in a step-by-step manner. f_a can be ignored after $\mathbf{F}_9$ is produced.

In testing whether a proper partial OF $F_i \cup \{g\} = \{f_1, f_2, .., f_{i+1}\}$ (where $g \in \mathbf{U}_{i+1}$) is canonical, we observe that the mappings must preserve $f_1 (= f_a)$. Thus we could use the set $\{\alpha : \alpha \in \mathbf{S}_{10}$ and $f_1^{\alpha} = f_1\}$, the cardinality of which is $2^5 5! = 3840$.

We implemented canonicity testing by mapping pairs of distinct one-factors. Similar to K_{10}, any pair of distinct one-factors of K_{10} - f forms either two disjoint cycles of lengths 4 and 6 (type '46') or a Hamiltonian cycle of length 10 (type '10'). The smallest one-factor in $\mathbf{U}_2$ that forms a type '46' structure with f_a is f_b= ((1, 3), (2, 4), (5, 7), (6, 9) (8, 10)), and the smallest one-factor in $\mathbf{U}_2$ that forms a type '10' structure is f_c= ((1, 3), (2, 5), (4, 7), (6, 9), (8, 10)). It follows then that the set of proper partial OF of rank 2, $\mathbf{F}_2$, is the set $\{(f_a, f_b), (f_a, f_c)\}$, where $f_a < f_b < f_c$.

To test the canonicity of $F_{i+1} = F_i \cup \{g\}$ (= $(f_1, f_2, \ldots, f_{i+1})$) at step i+1, we need only examine $f_1 f_j$, where $2 \leq j \leq i+1$, because f_1 (= f_a) must be fixed. Depending on $f_2 = f_b$ or $f_2 = f_c$, we have the following two cases:

(1) $f_1 f_2 = f_a f_b$ (type '46'):

We may ignore those $f_1 f_j$ of type '10', as mapping them into $f_a f_c$ would always make $F_{i+1}^{\alpha} > F_{i+1}$.

We map any f_1f_j, $2 \le j \le i+1$ of type '46' into f_af_b (in such a way that f_1 is mapped to f_a and f_j is mapped to f_b). To map into *any other* set of two one-factors of type '46' would always make $F_{i+1}{}^{\alpha} > F_{i+1}$. There are $(2\cdot2)\cdot(2\cdot3) = 24$ ways to do this. The maximum number of mappings for a F_{i+1} is $(i+1)\cdot24$.

(2) $f_1f_2 = f_af_c$ (type '10'):

All f_1f_j, $2 \le j \le i+1$ must be of type '10' (in general, no f_1f_j can be of a type corresponding to a canonical structure less than f_1f_2). Thus we discard those $g \in \mathbf{U}_{i+1}$ which form a type '46' structure with any of f_1, before the canonicity testing. We must map f_1 into f_a and f_j into f_c, where $2 \le j \le i+1$. Thus there are $(2\cdot5) = 10$ ways to map type '10' structures. The maximum number of mappings for F_{i+1} is $(i+1)\cdot10$.

7.4 One-factorizations of K_{10} - f

The number of canonical structures and CPU time required for each of the steps are listed in Table 7.1. The number of non-isomorphic OFs of K_{10} - f of types '46' and '10' are 2944 and 248 respectively. The algorithm required approximately 18 minutes of CPU time.

Table 7.1

Non-isomorphic canonical proper partial OFs of $K_{10} - f$

Step	# of canonical structures at step i+1			CPU time
i+1	type '46'	type '10'	total	(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

7.5 Howell designs H(8, 10)

The underlying graph of the Howell designs H(8, 10) is $K_{10} - f$. We used the orderly algorithms described in Section 4.4 to construct pairs of orthogonal OFs (F, G) of $K_{10} - f$. Let $F = (f_1, f_2, ..., f_9)$ and $G = (g_1, g_2, ..., g_9)$. Note that $g_1 = f_1 = f_a$. In testing for canonicity, we restrict the mappings to the following:

(1) Mappings for F. It suffices to examine those $\alpha \in \mathbf{S}_{10}$ such that $F^\alpha = F$ (and $f_1^\alpha = f_1$), since F is canonical. That is, we 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 $f_1f_2 = f_af_c$ (hence all f_1f_j are of type '10'), then all g_1g_j must necessarily be of type '10'.

(2) Mappings for G. There are two cases:

(a) There exists a g_1g_j of type '46', where $2 \le j \le 9$. We map all g_1g_j of type '46' into f_af_b (with g_1 mapped into f_a), 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 (with g_1 mapped into f_a).

In total, there are 18220 non-isomorphic (F, G) of K_{10} - f. It required 38 minutes of CPU time. Appendix 3 gives the frequency distribution of these designs, based on the number of non-isomorphic (F, G) for a given F.

7.6 Howell cubes and $H_4(8, 10)$

Using the algorithm outlined in Section 4.6, 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 Appendix 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 Appendix 4). This leaves us with 3 sets to which we apply canonicity testing. In this case, we simply use the α's in the group $\mathrm{Aut}(K_{10} - f) = \{\alpha : f_a^{\alpha} = f_a, \alpha \in \mathbf{S}_{10}\}$. We find one of them is canonical (set 2 in Appendix 4). In summary, we have

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

It is interesting to note that the set of four mutually orthogonal OFs can be constructed from a finite projective plane of order 8 [38].

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

7.7 Skew H(8, 10) designs

In [36], Lamken and Vanstone introduce skew Howell designs H(r, r + 2), where r is even, and give a construction for a skew H(4, 6). It is also reported that there does not exist a skew H(6, 8), and the first unsettled case was that of a skew H(8, 10). In this section, we perform an enumeration of skew H(8, 10), and we find that there are exactly three non-isomorphic examples.

A Howell design H(r, r + 2), say H, is said to be *skew* if there exist two symbols a, b, where {a, b} is not an edge of the underlying graph, such that the following properties are satisfied:

(1) Denote the r cells of H which contain a by T_a, and denote the r cells of H which contain b by T_b. Then $T_a \cup T_b$ consists of the r cells on the diagonal of H (say D), and r other cells which form a transversal of cells (say D') of H, such that D' is symmetric with respect to D (i.e. a cell $(i, j) \in D'$ if and only if cell $(j, i) \in D'$).

(2) Given any cell (i, j) $\notin D \cup D'$, precisely one of cell (i, j) and cell (j, i) is empty.

In Section 7.5, we enumerated all non-isomorphic H(8, 10); there are 18220 such Howell designs. It was therefore a straightforward test to see which of these designs could be written down in such a way that it forms a skew H(8, 10). This was done as follows. For any given H(8, 10), there are five possibilities for the pair {a, b}. For each possibility, the cells in $T_a \cup T_b$ form four 4-cycles (no matter how the Howell design is written down). For each 4-cycle, there are essentially two inequivalent ways of permuting the rows / columns containing the 4-cycle. There are thus only $2^5 = 32$ row / column permutations that must be considered (for each possible {a, b}).

As a result of these tests, we found precisely three non-isomorphic skew H(8, 10), which we record in Appendix 7.

CHAPTER 8

ENUMERATING ONE-FACTORIZATIONS AND HOWELL DESIGNS OF OTHER REGULAR GRAPHS

8.1 Introduction

In this chapter, we turn to the non-isomorphic OFs and Howell designs of several regular graphs of small order.

The non-isomorphic OFs and (i-dimensional) Howell designs have been enumerated (for all i) for all graphs on at most 10 vertices (see [7] , [51], and Chapter 7). It is not feasible to continue this enumeration to all graphs Gr on 12 vertices, for two reasons. If Gr is r-regular with r close to 12, the numbers $N_i(Gr)$ will be astronomical, and present techniques would not yield any results in a reasonable amount of time (see Chapter 5). If Gr is 6- or 7-regular, we can determine the numbers $N_i(Gr)$; the problem here is that there are too many graphs to test them all. In the remaining sections, we discuss the enumeration of OFs and sets of orthogonal OFs (that is, Howell designs) for several graphs on 10, 12 and 14 vertices.

Among our results are the following. From the twelve 6-regular graphs on 12 vertices having transitive automorphism groups, we found that there are precisely 24 non-isomorphic H(6, 12), and precisely one $H_3(6, 12)$. From the ten 7-regular graphs on 12 vertices having transitive automorphism groups, we found that there are precisely 1393 non-isomorphic H(7, 12), and precisely five

$H_3(7, 12)$. We also determined that there are exactly three $H^*(7, 12)$ designs. We found an example of an $H^{**}(13, 14)$, which was the smallest case of an $H^{**}(2n-1, 2n)$, and was not previously known to exist. Finally, we proved that there are precisely 2 non-isomorphic $H^{**}(9, 10)$.

8.2 6-regular graphs on 12 vertices

The case of 6-regular graphs on 12 vertices is particularly interesting, due to the non-existence of a pair of orthogonal Latin squares of order 6 (i.e. $N_2(K_{6,6}) = 0$). In [28], Hung and Mendelsohn presented the first example of an $H(6, 12)$. More recently, Brickell found a Howell cube $H_3(6, 12)$ for which the underlying graph is the icosahedron with antipodal points joined (see [11]). It is also worth mentioning that the automorphism group of this cube is the same as the automorphism group of the icosahedron (this group is isomorphic to $Z_2 \times A_5$).

In the hope of finding further examples, we investigated the 6-regular graphs on 12 vertices having a transitive automorphism group. There are precisely 12 such graphs (see [8]); we present a listing of the edges of the complements of these graphs in Appendix 8. From these 12 graphs, we found that there are precisely 24 non-isomorphic $H(6, 12)$, and precisely one $H_3(6, 12)$ (the Brickell cube). There are no examples of an $H_4(6, 12)$ in this class of graphs. A summary of our results is given in Table 8.1.

8.3 7-regular graphs on 12 vertices

As in Section 8.2, we looked at the graphs having transitive automorphism groups. For 7-regular graphs on 12 vertices, there are 10 such graphs (see [8]). We list the edges in the complements of these graphs in Appendix 9. From these 10 graphs, we found many more Howell designs: 1393 non-isomorphic H(7, 12), and five non-isomorphic $H_3(7, 12)$. The enumeration is summarized in Table 8.2. An example of an $H_3(7, 12)$ was not previously known; we present one of them in Appendix 10.

We also investigated two other 7-regular graphs on 12 vertices, namely, the graphs which correspond to the so-called **-designs.* Thus the underlying graph of H*(7, 12) has the form $K_5^c + Q_7$, where K_5^c is the complement of the complete graph on 5 vertices (hence it is a graph of 5 vertices with no edges), and Q_7 is either a 7-cycle or the disjoint union of a 3-cycle and a 4-cycle. In the first case, there are 4045 OFs but no H*(7, 12); in the second case, there are 1160 OFs and three non-isomorphic H*(7, 12), which are presented in Appendix 11. These are thus the smallest examples of H*(n, 2n-2) for n odd, since there are no Howell designs H(3, 4) or H(5, 8) (previously, the smallest example in this class was an H*(13, 24), constructed in [52]).

8.4 Algorithms

We modified the orderly algorithms for K_{2n} in Section 2.7 to enumerate the non-isomorphic OFs of the 6- and 7-regular graphs on 12 vertices in the preceding sections. We used the automorphism groups of these graphs in the canonicity testing, since their orders are fairly small (see Table 8.1). Depth-first algorithm without pruning was employed.

Orderly algorithms in Chapter 4 are used to enumerate the Howell designs of these graphs. Again, automorphism groups of the graphs are used to test the canonicity of sets of orthogonal OFs.

In total, enumeration for the 6-regular graphs took about 20 minutes of CPU time, while the 7-regular graphs required approximately 10 hours.

8.5 H**(13, 14)

Another special class of Howell designs are called ***-designs*. An H**(r, n) is defined to be an H(r, n) which satisfies the following two properties:

(1) there exists an (r - n / 2) x (r - n / 2) subarray of the Howell design which consists of empty cells,

(2) there exists a one-factor of the underlying graph which forms a transversal of the n / 2 rows and columns which do not meet the empty subarray of (1).

These may seem somewhat unusual properties to ask for, but it turns out that there is a powerful recursive construction for **-designs, which was instrumental in the proof of necessary and sufficient conditions for the existence of Room squares of side 2n+1 ($\neq$ 3 or 5); see [44].

There has recently been some interest in H**(2m-1, 2m) (that is, Room squares which are **-designs). Note that we can define an H**(2m-1, 2m) by requiring only that property (1) holds; property (2) then follows as a consequence. Such a design has several equivalent formulations, which are described in [62]: one of these is a partitioned balanced tournament design PBTD(m), and another is a pair of almost disjoint H(m, 2m). We elaborate on the second formulation. Two H(m, 2m), say D_1 and D_2 (on the same symbol set), having underlying graphs Gr_1 and Gr_2, respectively, are said to be *almost disjoint* if the following properties hold:

(1) $Gr_1 \cap Gr_2 = f$, where f is a one-factor

(2) $Gr_1 \cup Gr_2 - f = K_{2m}$, the complete graph on 2m vertices

(3) the edges of f occur in a row (or column) of D_1, and in a row (or column) of D_2.

$H^{**}(2m-1, 2m)$ do not exist for m = 2, 3, or 4 (see [62]). For $m \geq 5$, such a design is known to exist for all but 12 values of m ([37], [39] and [40]). The smallest unknown case was m = 7. We were able to construct two non-isomorphic examples of $H^{**}(13, 14)$, which we present in Appendix 12 as sets of almost disjoint H(7, 14).

These were found as follows. The H(7, 14) labelled D_1 was constructed by E. Lamken (private communication). Call the underlying graph Gr_1, and let f denote the one-factor occurring in the last column of D_1. We first enumerated all OFs of the graph $Gr_2 = (K_{14} - Gr_1) \cup f$ which contain f as a one-factor, using the orderly algorithms of Chapter 2. The automorphism group of Gr_2 (order = 5184) is used to test the canonicity of the OFs. There were precisely 5272 non-isomorphic OFs F of this type. For each such F, we determined all possible OFs G of Gr_2 orthogonal to F, such that G also contains f as a one-factor (see

Chapter 4). For only two of these 5272 OFs F could we find such a G orthogonal to F (up to isomorphism). The enumeration took 7 hours of CPU time.

8.6 H**(9, 10)

A H**(9, 10) is equivalent to a pair of almost disjoint H(5, 10), and an example has been constructed in [62] (see set 1 of Appendix 13). In this section, we generalized the approach used in the previous section, and carried out a complete enumeration of pairs of almost disjoint H(5, 10).

In [51], orthogonal OFs of all 5-regular graphs on 10 vertices (that is, H(5, 10)) were enumerated. In total, there are 5 graphs giving rise to a total of 6 H(5, 10)'s (see Table 9 in [51]). We number these graphs as in [51]: no. 2, no. 17, no. 50, no. 53 and no. 60. No. 60 is $K_{5,5}$ and admits 2 H(5, 10)'s. In [62], it was mentioned that a pair of almost disjoint H(5, 10) cannot have $K_{5,5}$ as one of the underlying graphs. ($K_{5,5}{}^c$ + f, where f is a one-factor on 10 vertices, is not isomorphic to any of these 5 graphs.) Thus we can restrict our investigation to **Gr** = {no. 2, no. 17, no. 50, no. 53}. We define an ordering on these graphs such that no. 2 < no. 17 < no. 50 < no. 53.

We now give the orderly algorithm for enumerating canonical pairs of almost disjoint H(5, 10). We remark that the algorithm can be modified easily for any complete graph of order 2n.

We use (D_1, D_2) to denote a pair of almost disjoint H(5, 10), where D_1 and D_2 are H(5, 10)'s of underlying graphs Gr_1 and Gr_2 respectively, and Gr_1 and Gr_2 are isomorphic to some graphs in **Gr**. Since we construct canonical

(D_1, D_2), D_1 must be canonical; that is, $D_1^{\alpha} \geq D_1$ for all $\alpha \in Aut(Gr_1)$. D_2 need not be canonical. Hence, $D_2 = (F, G)$, where $F < G$, and F, G are OFs of Gr_2 containing the special one-factor. The following is the pseudo-code for the algorithm.

```
FOR each Gr_1 ∈ Gr DO
    FOR each H(5, 10) D_1 having underlying graph Gr_1 DO
        FOR each one-factor f in D_1 DO {There are 10 possibilities.}
            Gr_2 = (K_10 - Gr_1) ∪ f;
            IF Gr_2 is isomorphic to some graph in Gr THEN
                IF Gr_1 ≤ Gr_2 THEN
                    construct all H(5, 10) D_2 having underlying graph Gr_2;
                    determine {π : f^π = f, and (Gr_1 - f)^π = (Gr_1 - f) or (Gr_2 - f)}
                    IF (D_1, D_2)^π < (D_1, D_2) for some π THEN
                        (D_1, D_2) is not canonical; discard it.
                    {Here, (D_1, D_2) is canonical.}
```

We implemented the algorithm above and found that there are 2 non-isomorphic pairs of almost disjoint H(5, 10). We list them in Appendix 13. It took about 3 minutes of CPU time.

Table 8.1

Howell designs from 6-regular graphs on 12 vertices having transitive automorphism groups

Graph No.	|Aut(Gr)|	DPM(Gr)	N(Gr)	N_2(Gr)	N_3(Gr)
1	768	368	190	0	0
2	144	348	469	3	0
3	48	344	1248	8	0
4	24	342	2018	0	0
5	96	392	1451	0	0
6	12	386	6932	1	0
7	120	368	733	4	1
8	12	354	4976	0	0
9	24	344	2216	5	0
10	48	344	1021	0	0
11	24	336	1983	3	0
12	1440	376	132	0	0

Notation: DPM(Gr) denotes the number of distinct one-factors of Gr.

Table 8.2

Howell designs from 7-regular graphs on 12 vertices having

transitive automorphism groups

Graph No.	|Aut(Gr)|	DPM(Gr)	N(Gr)	N_2(Gr)	N_3(Gr)
1	48	825	127222	84	1
2	24	837	270875	235	3
3	48	827	130176	103	0
4	48	824	130141	166	0
5	24	821	245138	189	0
6	24	808	218138	130	0
7	768	827	9145	47	0
8	144	820	43060	72	1
9	24	818	237042	264	0
10	48	804	110656	103	0

Notation: DPM(Gr) denotes the number of distinct one-factors of Gr.

CHAPTER 9

CONSTRUCTING PERFECT ONE-FACTORIZATIONS USING OTHER ALGORITHMS

9.1 Introduction

Most known examples of perfect OFs arise from starters or even starters. In [2], Anderson enumerates all perfect OFs in K_{2n} arising from starters and even starters, up to n = 11. These empirical results suggest that there exists a starter-induced perfect OF in K_{2n} for all $n \geq 6$, and an even starter-induced perfect OF in K_{2n} for all $n \geq 6$. Thus starters or even starters might provide new examples of perfect OF for larger values of n.

In this chapter, we construct starter-induced perfect OFs for K_{36} and K_{50}. The algorithms we use are hill-climbing and backtracking algorithms.

9.2 Starters and even starters

We need the following definitions for starters and even starters.

Let $\mathbf{Z}_m$ be the cyclic additive group on the set of m elements, {0, 1, ..., m-1}. A *starter* in $\mathbf{Z}_{2n-1}$ is a set $S = \{\{x_1, x_2\}, \{x_3, x_4\}, ..., \{x_{2n-3}, x_{2n-2}\}\}$ such that every non-zero element of $\mathbf{Z}_{2n-1}$ occurs as

(1) an element of some pair of S, and

(2) a difference of some pair of S.

Define $S^* = S \cup \{0, \infty\}$ and $\infty + g = g + \infty = \infty$ for all $g \in \mathbf{Z}_{2n-1}$. It is easy to see that $F = \{S^* + g : g \in \mathbf{Z}_{2n+1}\}$ is an OF of K_{2n} (see [43]).

An *even starter* in $\mathbf{Z}_{2n}$ is a set $E = \{\{x_1, x_2\}, \{x_3, x_4\}, \ldots \{x_{2n-3}, x_{2n-2}\}\}$ such that

(1) every non-zero element of $\mathbf{Z}_{2n}$ except one, denoted m, occurs as an element in some pair of E, and

(2) every non-zero element of $\mathbf{Z}_{2n}$ except n occurs as a difference of some pair of E.

Define $E^* = E \cup \{\{0, \infty_1\}\} \cup \{\{m, \infty_2\}\}$, and $g + \infty_i = \infty_i + g = \infty_i$ for $g \in \mathbf{Z}_{2n}$ and $i = 1, 2$. Also define $Q^* = \{\{g, g+n\} : g \in \mathbf{Z}_{2n}\} \cup \{\{\infty_1, \infty_2\}\}$. Then $F = \{E^* + g : g \in \mathbf{Z}_{2n}\} \cup \{Q^*\}$ is an OF of K_{2n+2} (see [43]).

9.3 Hill-climbing algorithms

Traditionally, backtracking algorithms have been used to construct designs on computer. However, the computer time required for these algorithms often grows exponentially with the order of the problems, making them impractical to produce designs of relatively smaller order. In these cases, hill-climbing algorithms often have more success. In fact, hill-climbing algorithms have been used in recent years to construct combinatorial designs such as strong starters, Steiner triple systems, Room squares and OFs, and to solve many other optimization problems (see [20], [45], [59], [61], [63] and [64]). Researchers have found that this approach works very well for certain problems.

Hill-climbing is a non-enumerative algorithm which constructs designs in a non-deterministic manner using some heuristics. It implicitly assumes the existence of a solution. Hence in order for hill-climbing to be successful, there must be a solution, or better still, many solutions. The heuristics used in the algorithm to build up the design should be fast, as generally many trials (repetition of these heuristics) are needed to successfully construct a design.

We use a modification of the hill-climbing algorithm in [20] to generate (even) starters. For each (even) starter generated, we then test the induced OF for perfection. We now give a brief description of the hill-climbing algorithm used.

Define a *partial starter* to be a set $S' = \{\{x_1, x_2\}, \{x_3, x_4\}, \ldots, \{x_{2m-3}, x_{2m-2}\}\}$, where $m \leq n$, satisfying the conditions that (1) x_i's are distinct non-zero elements of $\mathbf{Z}_{2n-1}$, and (2) $(x_{2i-1} - x_{2i}) \neq \pm(x_{2j-1}, x_{2j})$ for $i \neq j$. Similarly, we define a *partial even starter* to be a set $E' = \{\{x_1, x_2\}, \{x_3, x_4\}, \ldots, \{x_{2m-3}, x_{2m-2}\}\}$, where $m \leq n$, satisfying the conditions (1) x_i's are distinct non-zero elements of $\mathbf{Z}_{2n}$, and (2) $(x_{2i-1} - x_{2i}) \neq \pm(x_{2j-1}, x_{2j})$ for $i \neq j$. Note that when $m = n$, we have either a (complete) starter S or a (complete) even starter E.

The algorithm non-deterministically constructs the pairs in the (even) starter using one of two possible heuristics. At a given stage in the algorithm, we have a partial (even) starter S' (E'). We say that an element or difference is *used* or *unused* depending on whether it occurs in the (current) partial (even) starter.

(1) An unused element u and an unused difference d are picked randomly. This determines a second element v of the pair (either of u + d or u - d). If v is unused, then add the pair {u, v} to the partial (even) starter S' (E'). Otherwise, delete the pair containing v from the partial (even) starter S' (E'), and add the pair {u, v}.

(2) Choose two unused elements (u and v, where u < v). If the difference d (d = v - u) is unused, then add the pair {u, v}. Otherwise, delete the pair that has the difference d from the partial (even) starter S' (E'), and add the pair {u, v}.

Note that at no time does the number of pairs in the partial (even) starter decrease. Although we cannot guarantee that (even) starters will always be found by this algorithm, in actuality it seems to work all the time, and it is very fast. For those readers interested in this class of algorithms, we suggest [61].

9.4 A perfect one-factorization of K_{36}

We implemented the hill-climbing algorithm for both starters and even starters. After 15 hours of CPU time (having constructed a total of 6 million starters and even starters) we found the following starter in $\mathbf{Z}_{35}$ which induces a perfect OF:

{ {14, 15}, {5, 7}, {19, 22}, {28, 32}, {25, 30}, {11, 17}, {6, 13}, {18, 26}, {29, 3}, {34, 9}, {20, 31}, {33, 10}, {23, 1}, {2, 16}, {12, 27}, {8, 24}, {4, 21} }.

The automorphism group of the induced perfect OF is Z_{35}.

We were not as lucky with K_{40}, having spent about 100 hours of CPU time without finding a perfect OF.

9.5 Statistics on hill-climbing algorithms

The probability of finding a perfect OF by means of the hill-climbing algorithm described in Section 9.3 depends on two factors.

1) What is the probability that a random (even) starter-induced OF is perfect?
2) Does the hill-climbing algorithm generate random (even) starter-induced OFs?

To help answer these two questions, we performed some experiments on K_{2n} (for $16 \le 2n \le 30$). For $2n \le 22$, a complete enumeration of (even) starters was done in [2]. By testing the resulting OFs for perfection, we obtain exact probabilities for 1), dividing the number of (even) starters into the number of (even) starter-induced perfect OFs. Due to the large number of (even) starters for $2n > 22$, it is computationally infeasible to extend this enumeration to larger orders.

To help answer 2), we generated many (even) starters using the hill-climbing algorithm in order to estimate the probability that a given (even) starter produced by the hill-climbing algorithm induces a perfect OF. (We note that the perfect OFs generated are not necessarily non-isomorphic, nor even distinct.) These results are summarized in Tables 9.1 and 9.2. The two sets of probabilities (for $16 \leq 2n \leq 22$) appear to be fairly close, suggesting that the (even) starters generated by the hill-climbing algorithm are random.

Define S(n) to be the expected number of (even) starters required to obtain a perfect OF on K_n. From our empirical evidence, it appears that $\log_{10} S(n)$ is a linear function of n (i.e. S(n) increases exponentially as a function of n). Using a linear least-squares approximation against the sample data, we estimate, for the case of starters, that $S(n) \approx 10^{.288n - 2.843}$; and for even starters, that $S(n) \approx 10^{.229n - 1.977}$. Substituting n = 40, we obtain estimates of $10^{8.7}$ and $10^{7.2}$ respectively. That is, we would expect to have to generate over 15,000,000 even starters before we would expect to find a perfect OF for K_{40}, and even more starters.

The computer results provide further empirical evidence that perfect OFs are very difficult to construct. Given enough computer time, we might find a perfect OF for K_{40}, but these techniques will most likely be unsuccesful for larger orders.

9.6 A perfect one-factorization of K_{50}

The computer results in the previous sections suggest that, to construct (even) starter-induced perfect OFs of complete graphs of orders larger than 36, we would probably have to try a different algorithm. Alternatively we may restrict ourselves to (even) starters that have additional structures, so as to cut down on the computer search time. We tried the second approach and succeeded in finding a perfect OF for K_{50}.

Recently, Ihrig [30] showed that if 2n-1 is not prime, then the order of the automorphism group of a starter-induced perfect OF of K_{2n} must be odd and divide $(2n-1)\cdot GCD(\varphi(2n-1),n-1)$, where φ is the Euler function. In fact, the "maximum" automorphism group **A** of the perfect OF is a semidirect product of (i) $\mathbf{Z}_{2n-1}$ and (ii) a subgroup of the multiplicative group of units in $\mathbf{Z}_{2n-1}$, for which the order is odd and divides n-1 (hence the order of **A** is at most $(2n-1)\cdot GCD(\varphi(2n-1), n-1)$).

In [2], Anderson enumerated the starter-induced perfect OF of K_{2n} (for n up to 11). The results indicated that there exists a starter-induced perfect OF the automorphism group of which has the largest permissible order.

Often, $GCD(\varphi(2n-1), n-1) = 1$, which means that the automorphism group of the starter-induced perfect OF is simply $\mathbf{Z}_{2n-1}$. The smallest order of K_{2n} for which the existence of perfect OF is unknown, and for which the largest odd factor in $GCD(\varphi(2n-1), n-1) > 1$, is $2n = 50$. Here, the largest odd factor in $GCD(\varphi(49), 24) = 3$, and thus the largest permissible order of the automorphism group of a perfect OF of K_{50} is 49x3.

Ihrig suggested there could exist a starter in the ring $\mathbf{Z}_{49}$ which is fixed by the multiplicative subgroup {1, 18, 30} and which generates a perfect OF of order 50. Thus this perfect OF will have the semidirect product of $\mathbf{Z}_{49}$ with $\mathbf{Z}_3$ as its automorphism group. We carried out an exhaustive search using backtracking, and found that there are precisely 938 such starters, 67 of which are non-isomorphic. The enumeration took about 5 hours of CPU time. A starter is as follows:

1	2		30	11		18	36
4	6		22	33		23	10
42	45		35	27		21	26
12	16		17	39		20	43
32	38		29	13		37	47
8	15		44	9		46	25
19	28		31	7		48	14
40	3		24	41		34	5

9.7 Backtracking algorithm

To explain the backtracking algorithm used in finding the starter for $\mathbf{Z}_{50}$, we need the following definitions.

Let **D** be the set of differences of the n-1 pairs of a starter S of $\mathbf{Z}_{2n-1}$; that is, $\mathbf{D} = \{\pm d : d \in \mathbf{Z}_n \backslash \{0\}\}$. Let **M** be a multiplicative subgroup of $\mathbf{Z}_{2n-1}$. We observe that for a given $\pm d$, the set $\{\pm md : m \in \mathbf{M}\}$ is a subset of **D** and forms an orbit. That is, **M** partitions the differences of **D** into disjoint orbits. We denote these orbits by $\mathbf{O} = \{O_1, O_2, ..., O_m\}$, where O_i is a subset of **D** and $O_i \cap O_j = \emptyset$ whenever $i \neq j$. Let $\text{rep}(O_i) = \pm d$, where d is the smallest element in the set $\{d : \pm d \in O_i\}$. Define rep(**O**) to be the set $\{\text{rep}(O_i) : O_i \in \mathbf{O}\}$.

We note that adding a pair {a, b} to a partial starter causes all the pairs in the set $\{\{ma, mb\} : m \in \mathbf{M}\}$ to be added, since **M** fixes the starter. Consequently, adding a pair {a, b} with the differences $\pm(a-b) \in O_i$ to a partial starter implies that all other differences in O_i are also used in the partial starter. Thus, in building up a starter, it suffices to consider only the differences in rep(**O**).

The backtracking algorithm constructs the set $\mathbf{T}$ of $|\text{rep}(\mathbf{O})|$ pairs of elements $\{a, b\}$, where $a, b \in \mathbf{Z}_{2n-1} \setminus \{0\}$, such that (i) their differences are distinct and are in the set $\text{rep}(\mathbf{O})$, and (ii) the set $\{ma, mb : m \in \mathbf{M} \text{ and } \{a, b\} \in \mathbf{T}\}$ is identical to $\mathbf{Z}_{2n-1} \setminus \{0\}$. Once this is done, the starter S is just simply the set $\{\{ma, mb\} : m \in \mathbf{M} \text{ and } \{a, b\} \in \mathbf{T}\}$.

Thus for K_{50}, we have $\mathbf{D} = \{1, 2, ..., 24\}$, $\mathbf{M} = \{1, 18, 30\}$, $\mathbf{O} = \{\{\pm 1, \pm 19, \pm 18\}, \{\pm 2, \pm 11, \pm 13\}, \{\pm 3, \pm 8, \pm 5\}, \{\pm 4, \pm 22, \pm 23\}, \{\pm 6, \pm 16, \pm 10\}, \{\pm 7, \pm 12, \pm 21\}, \{\pm 9, \pm 24, \pm 15\}, \{\pm 12, \pm 17, \pm 20\}\}$, and $\text{rep}(\mathbf{O}) = \{\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 7, \pm 9, \pm 12\}$. Note that the set $\mathbf{T}$ would have 8 pairs eventually.

The following recursive pseudo-code describes the backtracking algorithm that can be used to construct all starters of K_{2n-1} that is fixed by the multiplicative subgroup $\mathbf{M}$. To invoke the algorithm, use Extend(∅, ∅, 1).

```
PROCEDURE Extend (T, S, i)

IF i > |rep(O)| THEN
   {we have the starter S,} check the perfectness of the induced OF
ELSE
   FOR each j ∈ Z_{2n-1} \ {0} DO
      IF j is unused THEN {Here j is not in S.}
         k := j + rep(O)_i;
         IF k ≠ 0 and is unused THEN {Here k is valid and is not in S.}
            W := ∅;
            FOR each m ∈ M DO W := W ∪ {mj, mk};
```

IF $W \cap S = \varnothing$ THEN

$\mathbf{T} := \mathbf{T} \cup \{j, k\}$;

$S := S \cup \mathbf{W}$;

extend($\mathbf{T}$, S, i+1);

$\mathbf{T} := \mathbf{T} - \{j, k\}$;

$S := S - \mathbf{W}$.

9.8 Results on K_{92}

The next unknown case where GCD(φ(2n-1), n-1) has an odd factor exceeding 1 is 2n=92. Here, GCD(φ(91), 45) = 9. The "maximum" automorphism group would be a semidirect product of $\mathbf{Z}_{91}$ with $\mathbf{Z}_3 \times \mathbf{Z}_3$. We conducted an exhaustive search for starters in the ring $\mathbf{Z}_{91}$ which is fixed by the product of the multiplicative subgroup {1, 2, 4} of $\mathbf{Z}_7$ and the multiplicative subgroup {1, 3, 9} of $\mathbf{Z}_{13}$ (which corresponds to {1, 9, 16, 22, 29, 53, 74, 79, 81} of $\mathbf{Z}_{91}$). After approximately 12 minutes of computer time, we found no such starter that generates a perfect OF.

Here, we have $\mathbf{D} = \{1, 2, ..., 45\}$, $\mathbf{M} = \{1, 9, 16, 22, 29, 53, 74, 79, 81\}$, $\mathbf{O} = \{\{\pm1, \pm9, \pm16, \pm22, \pm29, \pm38, \pm17, \pm12, \pm10\}, \{\pm2, \pm18, \pm32, \pm44, \pm33, \pm15, \pm34, \pm24, \pm20\}, \{\pm3, \pm27, \pm43, \pm25, \pm4, \pm23, \pm40, \pm36, \pm30\}, \{\pm5, \pm45, \pm11, \pm19, \pm37, \pm8, \pm6, \pm31, \pm41\}, \{\pm7, \pm21, \pm28\}, \{\pm13, \pm26, \pm39\}, \{\pm14, \pm35, \pm42\}\}$, and rep($\mathbf{O}$) $= \{\pm1, \pm2, \pm3, \pm5, \pm7, \pm13, \pm14\}$. Note that the last 3 orbits in $\mathbf{O}$ are shorter than the other.

For composite values of 2n-1, this is the first example where there does not exist a perfect OF for which the automorphism group has order equal to the product of (2n-1) and the largest odd factor in GCD(φ(2n-1), n-1).

A complete search for starters in the semidirect product of $\mathbf{Z}_{91}$ with $\mathbf{Z}_3$ looks quite impossible at this time.

Table 9.1

Statistics for starter-induced perfect OFs

Graph	Hill-climbing algorithm			Exhaustive enumeration		
	No of starters	No of perfect OF	Estimated prob of perfect OF	No of starters	No of perfect OF	True prob of perfect OF
K_{16}	137000	1851	0.135×10^{-1}	631	8	0.127×10^{-1}
K_{18}	122000	526	0.432×10^{-2}	3857	17	0.441×10^{-2}
K_{20}	106000	284	0.267×10^{-2}	25905	65	0.251×10^{-2}
K_{22}	399000	84	0.209×10^{-3}	188181	36	0.191×10^{-3}
K_{24}	499000	37	0.741×10^{-4}			
K_{26}	2102000	72	0.343×10^{-4}			
K_{28}	2463000	13	0.528×10^{-5}			
K_{30}	2638000	4	0.152×10^{-5}			

Table 9.2

Statistics for even starter-induced perfect OFs

Graph	Hill-climbing algorithm			Exhaustive enumeration		
	No of even starters	No of perfect OF	Estimated prob of perfect OF	No of even starters	No of perfect OF	True prob of perfect OF
K_{16}	351000	4490	0.128×10^{-1}	960	12	0.125×10^{-1}
K_{18}	624000	8371	0.134×10^{-1}	5760	80	0.139×10^{-1}
K_{20}	546000	1476	0.270×10^{-2}	42816	120	0.280×10^{-2}
K_{22}	475000	412	0.869×10^{-3}	320512	272	0.849×10^{-3}
K_{24}	423000	86	0.203×10^{-3}			
K_{26}	394000	44	0.112×10^{-3}			
K_{28}	1135000	40	0.352×10^{-4}			
K_{30}	5596000	75	0.134×10^{-4}			

CHAPTER 10

CONCLUSION

10.1 Summary

In this thesis, we defined the canonicity concept, and developed various orderly algorithms to enumerate canonical (non-isomorphic) OFs and Howell designs of regular graphs.

These algorithms worked fairly well for regular graphs with small automorphism groups. We were able to carry out complete enumerations of canonical OFs and Howell designs of K_{10} minus a one-factor, and for all 6- and 7-regular graphs on 12 vertices containing transitive automorphism groups.

For regular graphs with large automorphism groups (for example, complete graphs), orderly algorithms have not been successful in completely enumerating the OFs of K_{12} and perfect OFs of K_{14}, due to the large amount of CPU time required. Consequently, we turned to orderly algorithms that enumerate canonical OFs containing prescribed automorphism groups. Using these algorithms, we were able to enumerate all canonical perfect OFs of K_{14} containing non-trivial automorphism groups. We also enumerated all OFs of K_{12} containing non-trivial automorphism groups (except those containing exactly one automorphism of order 2).

Special classes of Howell designs for several graphs were enumerated by modifying the orderly algorithms. These classes include Skew designs,

*- and **-designs.

In an attempt to find perfect OFs of complete graphs of larger orders, we used hill-climbing and backtracking algorithms to construct (even) starter-induced perfect OFs. We succeeded in finding examples of perfect OFs for K_{36} and K_{50}.

10.2 Open problems

There remain many interesting open problems.

The complete enumeration of non-isomorphic OFs of K_{12} is still not resolved. Using the algorithms in this thesis, OFs of K_{12} containing exactly one automorphism of order 2 can probably be enumerated in less than 100 hours of CPU time. However, enumeration of automorphism-free OFs of K_{12} will require a lot more time.

Similar comments also apply to the enumeration of perfect OFs of K_{14}: it will probably take many many hours of CPU time before the entire problem can be resolved. Since the existence of an automorphism-free perfect OF for K_{2n} remains an open problem, it would be interesting to see whether there exists an automorphism-free perfect OF for K_{14}.

We would like to comment that the current difficulty in carrying out complete enumerations for OFs of K_{12} and P1Fs of K_{14} is due mainly to the complexity of testing canonicity, and to the fact that there are many non-isomorphic (partial) structures. In fact, for complete graphs of larger order, a complete enumeration of non-isomorphic OFs with the orderly algorithms of Chapter 2 is computationally intractible. In these cases, "automorphism orderly algorithms" as described in Chapter 3 may be used to enumerate OFs containing certain automorphism groups (for example, cyclic OF).

It was shown in Chapter 6 that a perfect OF containing a P element does not exist for K_{2n} when n (> 2) is even. The question of existence of such OF for K_{2n} when n is odd appears to be difficult.

We would like to comment that the concept of canonicity and orderly algorithms can also apply to other combinatorial design problems, such as enumeration of non-isomorphic balanced incomplete block designs, BIBD(v, k, λ), containing certain automorphism groups. It seems hopeful that these algorithms will be fruitful in obtaining new results for other combinatorial design problems.

Existence of perfect OFs for all complete graphs remains an open, difficult problem. The smallest unknown case is now K_{40}. Empirical statistics with hill-climbing algorithms suggest that the expected number of (even) starters required to obtain a (even) starter-induced perfect OF of K_{2n} increases exponentially as a function of n. Thus, although we did find an example of starter-induced perfect OF of K_{36} using the hill climbing algorithms, it appears that these techniques will likely not be successful for larger orders. Algorithms other than the current hill-climbing algorithms will probably be needed to find an example for K_{40}.

Given an r-regular graph Gr on 2n vertices, it is a well-known conjecture that there exists an OF of Gr if $r \geq n$. The best result so far was obtained by Chetwynd and Hilton [15], who showed that this conjecture is true if $r \geq (6/7) \cdot 2n$. Whether a Chetwynd-Hilton type of result holds for Howell designs remains an open question. However, it should be noted that for r = n, n+1 and n+2, there are examples of graphs Gr for which OFs exist but Howell designs do not exist (see Tables 8.1 and 8.2, and Section 1.4). Infinite families of graphs for which Howell designs do not exist are not known at present.

APPENDIX 1

Cycle structures of admissible automorphisms of OFs of K_{12}

Case No.	Cycle Structure	Eliminated by Lemma	Case No.	Cycle Structure	Eliminated by Lemma
1	12^1		2	$11^1 1^1$	
3	$10^1 2^1$		4	$10^1 1^2$	
5	$9^1 3^1$	6.5	6	$9^1 2^1 1^1$	6.8
7	$9^1 1^3$	6.2	8	$8^1 4^1$	
9	$8^1 3^1 1^1$	6.8	10	$8^1 2^2$	
11	$8^1 2^1 1^2$		12	$8^1 1^4$	6.6
13	$7^1 5^1$	6.8	14	$7^1 4^1 1^1$	6.8
15	$7^1 3^1 2^1$	6.5	16	$7^1 3^1 1^2$	6.8
17	$7^1 2^2 1^1$	6.7	18	$7^1 2^1 1^3$	6.2
19	$7^1 1^5$	6.2	20	6^2	
21	$6^1 5^1 1^1$	6.8	22	$6^1 4^1 2^1$	6.8
23	$6^1 4^1 1^2$	6.8	24	$6^1 3^2$	a^3 (case 74)
25	$6^1 3^1 2^1 1^1$	6.7	26	$6^1 3^1 1^3$	6.2
27	$6^1 2^3$	a^2 (case 65)	28	$6^1 2^2 1^2$	a^2 (case 65)
29	$6^1 2^1 1^4$	a^2 (case 65)	30	$6^1 1^6$	6.4
31	$5^2 2^1$	a^5 (case 76)	32	$5^2 1^2$	
33	$5^1 4^1 3^1$	6.5	34	$5^1 4^1 2^1 1^1$	6.7
35	$5^1 4^1 1^3$	6.2	36	$5^1 3^2 1^1$	6.8
37	$5^1 3^1 2^2$	6.5	38	$5^1 3^1 2^1 1^2$	6.8

39	$5^1 3^1 1^4$	6.8	40	$5^1 2^3 1^1$	6.7
41	$5^1 2^2 1^3$	6.2	42	$5^1 2^1 1^5$	6.2
43	$5^1 1^7$	6.2	44	4^3	
45	$4^2 3^1 1^1$	6.8	46	$4^2 2^2$	
47	$4^2 2^1 1^2$		48	$4^2 1^4$	
49	$4^1 3^2 2^1$	6.8	50	$4^1 3^2 1^2$	6.8
51	$4^1 3^1 2^2 1^1$	6.7	52	$4^1 3^1 2^1 1^3$	6.2
53	$4^1 3^1 1^5$	6.2	54	$4^1 2^4$	a^2 (case 75)
55	$4^1 2^3 1^2$	a^2 (case 75)	56	$4^1 2^2 1^4$	a^2 (case 75)
57	$4^1 2^1 1^6$	6.4	58	$4^1 1^8$	6.3
59	3^4		60	$3^3 2^1 1^1$	6.7
61	$3^3 1^3$	6.2	62	$3^2 2^3$	a^3 (case 74)
63	$3^2 2^2 1^2$	a^3 (case 75)	64	$3^2 2^1 1^4$	a^3 (case 76)
65	$3^2 1^6$	6.4	66	$3^1 2^4 1^1$	6.7
67	$3^1 2^3 1^3$	6.2	68	$3^1 2^2 1^5$	6.2
69	$3^1 2^1 1^7$	6.2	70	$3^1 1^9$	6.2
71	2^6		72	$2^5 1^2$	
73	$2^4 1^4$		74	$2^3 1^6$	6.4
75	$2^2 1^8$	6.3	76	$2^1 1^{10}$	6.3
77	1^{12}				

APPENDIX 2

Perfect OFs of K_{14} and their automorphism groups

Set 1: $|A| = 84$ (GA_{14})

$A = \langle g1, g2 \rangle$

$g1 = (1\ 12\ 11\ 8\ 3\ 7)(2\ 14\ 13\ 10\ 5\ 9)$

$g2 = (1\ 13\ 4\ 9\ 8\ 5\ 12\ 2\ 11\ 6\ 7\ 10\ 3\ 14)$

g1 induces $(f_2\ f_4\ f_8\ f_{12}\ f_{11}\ f_7\ f_3)(f_5\ f_6)(f_9\ f_{10})(f_{13}\ f_{14})$

g2 induces $(f_2\ f_3\ f_4\ f_{11}\ f_7\ f_{12})(f_5\ f_{10}\ f_{14}\ f_6\ f_9\ f_{13})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 8	7 13	10 12	11 14
1 6	2 3	4 10	5 7	8 14	9 11	12 13
1 7	2 9	3 5	4 11	6 13	8 12	10 14
1 8	2 10	3 12	4 6	5 14	7 11	9 13
1 9	2 8	3 13	4 5	6 12	7 14	10 11
1 10	2 7	3 6	4 14	5 11	8 13	9 12
1 11	2 13	3 7	4 12	5 9	6 14	8 10
1 12	2 14	3 11	4 8	5 13	6 10	7 9
1 13	2 12	3 14	4 9	5 8	6 11	7 10
1 14	2 11	3 10	4 13	5 12	6 7	8 9

Set 2: $|A| = 4$

$A \cong \mathbb{Z}_2 \times \mathbb{Z}_2$

$A = \langle g1, g2 \rangle$

g1 = (1 2)(3 5)(4 6)(7 9)(8 10)(11 13)(12 14)

g2 = (1 2)(3 6)(4 5)(7 10)(8 9)(11 14)(12 13)

g1 induces $(f_5\ \ f_6)(f_9\ \ f_{10})(f_{13}\ \ f_{14})$

g2 induces $(f_3\ \ f_4)(f_7\ \ f_8)(f_{11}\ \ f_{12})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 8	7 13	10 12	11 14
1 6	2 3	4 10	5 7	8 14	9 11	12 13
1 7	2 9	3 13	4 6	5 11	8 12	10 14
1 8	2 10	3 5	4 14	6 12	7 11	9 13
1 9	2 8	3 12	4 5	6 13	7 14	10 11
1 10	2 7	3 6	4 11	5 14	8 13	9 12
1 11	2 13	3 14	4 8	5 12	6 10	7 9
1 12	2 14	3 7	4 13	5 9	6 11	8 10
1 13	2 12	3 11	4 9	5 8	6 14	7 10
1 14	2 11	3 10	4 12	5 13	6 7	8 9

Set 3: $|A| = 12.$

$A \cong \mathbf{Z}_2 \times \mathbf{Z}_6$

A = <g1, g2>

g1 = (3 11 10 4 12 9)(5 13 8 6 14 7)

g2 = (1 2)(3 6)(4 5)(7 10)(8 9)(11 14)(12 13)

g1 induces $(f_3\ f_{11}\ f_{10}\ f_4\ f_{12}\ f_9)$

$(f_5\ f_{13}\ f_8\ f_6\ f_{14}\ f_7)$

g2 induces $(f_3\ f_4)(f_{11}\ f_{12})(f_9\ f_{10})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 8	7 13	10 12	11 14
1 6	2 3	4 10	5 7	8 14	9 11	12 13
1 7	2 10	3 6	4 11	5 14	8 13	9 12
1 8	2 9	3 12	4 5	6 13	7 14	10 11
1 9	2 7	3 13	4 6	5 11	8 12	10 14
1 10	2 8	3 5	4 14	6 12	7 11	9 13
1 11	2 13	3 14	4 8	5 12	6 10	7 9
1 12	2 14	3 7	4 13	5 9	6 11	8 10
1 13	2 12	3 11	4 9	5 8	6 14	7 10
1 14	2 11	3 10	4 12	5 13	6 7	8 9

Set 4: $|A| = 6.$

$A \cong \mathbb{Z}_6$

A = <g1>

g1 = (3 10 7 4 9 8)(5 14 12 6 13 11)

g1 induces $(f_3 \quad f_{10} \quad f_7 \quad f_4 \quad f_9 \quad f_8)$

$(f_5 \quad f_{14} \quad f_{12} \quad f_6 \quad f_{13} \quad f_{11})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 8	7 13	10 12	11 14
1 6	2 3	4 10	5 7	8 14	9 11	12 13
1 7	2 12	3 11	4 5	6 13	8 9	10 14
1 8	2 11	3 6	4 12	5 14	7 10	9 13
1 9	2 13	3 10	4 6	5 11	7 14	8 12
1 10	2 14	3 5	4 9	6 12	7 11	8 13
1 11	2 7	3 14	4 8	5 13	6 10	9 12
1 12	2 8	3 7	4 13	5 9	6 14	10 11
1 13	2 10	3 12	4 14	5 8	6 11	7 9
1 14	2 9	3 13	4 11	5 12	6 7	8 10

Set 5: $|A| = 6$

$A \cong \mathbb{Z}_6$

$A = \langle g1 \rangle$

$g1 = (3\ 10\ 7\ 4\ 9\ 8)(5\ 13\ 11\ 6\ 14\ 12)$

g1 induces $(f_3\ \ f_{10}\ \ f_7\ \ f_4\ \ f_9\ \ f_8)$

$(f_5\ \ f_{13}\ \ f_{11}\ \ f_6\ \ f_{14}\ \ f_{12})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 8	7 14	10 11	12 13
1 6	2 3	4 10	5 7	8 13	9 12	11 14
1 7	2 11	3 12	4 6	5 13	8 9	10 14
1 8	2 12	3 5	4 11	6 14	7 10	9 13
1 9	2 14	3 10	4 5	6 11	7 13	8 12
1 10	2 13	3 6	4 9	5 12	7 11	8 14
1 11	2 8	3 7	4 14	5 9	6 13	10 12
1 12	2 7	3 13	4 8	5 14	6 10	9 11
1 13	2 9	3 14	4 12	5 11	6 7	8 10
1 14	2 10	3 11	4 13	5 8	6 12	7 9

Set 6: $|A| = 6$

$A \cong \mathbf{Z}_6$

$A = \langle g1 \rangle$

g1 = (3 14 12 4 13 11)(5 9 8 6 10 7)

g1 induces $(f_3\ \ f_{14}\ \ f_{12}\ \ f_4\ \ f_{13}\ \ f_{11})$

$(f_5\ \ f_9\ \ f_8\ \ f_6\ \ f_{10}\ \ f_7)$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 8	7 14	10 11	12 13
1 6	2 3	4 10	5 7	8 13	9 12	11 14
1 7	2 12	3 10	4 14	5 11	6 13	8 9
1 8	2 11	3 13	4 9	5 14	6 12	7 10
1 9	2 13	3 7	4 11	5 12	6 10	8 14
1 10	2 14	3 12	4 8	5 9	6 11	7 13
1 11	2 7	3 14	4 6	5 8	9 13	10 12
1 12	2 8	3 5	4 13	6 7	9 11	10 14
1 13	2 10	3 11	4 5	6 14	7 9	8 12
1 14	2 9	3 6	4 12	5 13	7 11	8 10

Set 7: $|A| = 2$

$A \cong \mathbb{Z}_2$

$A = \langle g1 \rangle$

$g1 = (3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12)(13\ 14)$

$g1$ induces $(f_3\ \ f_4)(f_5\ \ f_6)(f_7\ \ f_8)$

$(f_9\ \ f_{10})(f_{11}\ \ f_{12})(f_{13}\ \ f_{14})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 14	7 13	8 12	10 11
1 6	2 3	4 10	5 13	7 11	8 14	9 12
1 7	2 12	3 6	4 13	5 11	8 9	10 14
1 8	2 11	3 14	4 5	6 12	7 10	9 13
1 9	2 8	3 7	4 11	5 14	6 10	12 13
1 10	2 7	3 12	4 8	5 9	6 13	11 14
1 11	2 13	3 10	4 14	5 12	6 8	7 9
1 12	2 14	3 13	4 9	5 7	6 11	8 10
1 13	2 9	3 11	4 6	5 8	7 14	10 12
1 14	2 10	3 5	4 12	6 7	8 13	9 11

Set 8: $|A| = 2$

$A \cong \mathbb{Z}_2$

$A = \langle g1 \rangle$

$g1 = (3\ 4)(5\ 6)(7\ 8)(9\ 10)(11\ 12)(13\ 14)$

$g1$ induces $(f_3\ \ f_4)(f_5\ \ f_6)(f_7\ \ f_8)$

$(f_9\ \ f_{10})(f_{11}\ \ f_{12})(f_{13}\ \ f_{14})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 11	6 13	7 9	8 12	10 14
1 6	2 3	4 12	5 14	7 11	8 10	9 13
1 7	2 9	3 6	4 13	5 8	10 12	11 14
1 8	2 10	3 14	4 5	6 7	9 11	12 13
1 9	2 11	3 10	4 8	5 12	6 14	7 13
1 10	2 12	3 7	4 9	5 13	6 11	8 14
1 11	2 14	3 12	4 6	5 9	7 10	8 13
1 12	2 13	3 5	4 11	6 10	7 14	8 9
1 13	2 8	3 9	4 14	5 7	6 12	10 11
1 14	2 7	3 13	4 10	5 11	6 8	9 12

Set 9: $|A| = 12$

$A \cong Q_6$ (dicyclic group)

$A = \langle g1, g2 \rangle$

g1 = (3 5 4 6)(7 13 8 14)(9 11 10 12)

g2 = (3 13 10 4 14 9)(5 11 7 6 12 8)

g1 induces $(f_3\ f_5\ f_4\ f_6)(f_7\ f_{13}\ f_8\ f_{14})$
$(f_9\ f_{11}\ f_{10}\ f_{12})$

g2 induces $(f_3\ f_{13}\ f_{10}\ f_4\ f_{14}\ f_9)$
$(f_5\ f_{11}\ f_7\ f_6\ f_{12}\ f_8)$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 11	6 13	7 9	8 12	10 14
1 6	2 3	4 12	5 14	7 11	8 10	9 13
1 7	2 9	3 14	4 8	5 11	6 10	12 13
1 8	2 10	3 7	4 13	5 9	6 12	11 14
1 9	2 8	3 13	4 6	5 12	7 14	10 11
1 10	2 7	3 5	4 14	6 11	8 13	9 12
1 11	2 14	3 6	4 9	5 8	7 13	10 12
1 12	2 13	3 10	4 5	6 7	8 14	9 11
1 13	2 11	3 12	4 10	5 7	6 14	8 9
1 14	2 12	3 9	4 11	5 13	6 8	7 10

Set 10: $|A| = 12$

$A \cong Q_6$ (dicyclic group)

$A = \langle g1, g2 \rangle$

g1 = (3 5 4 6)(7 12 8 11)(9 13 10 14)

g2 = (3 7 14 4 8 13)(5 10 11 6 9 12)

g1 induces $(f_3\ f_5\ f_4\ f_6)(f_7\ f_{12}\ f_8\ f_{11})$
$(f_9\ f_{13}\ f_{10}\ f_{14})$

g2 induces $(f_3\ f_7\ f_{14}\ f_4\ f_8\ f_{13})$
$(f_5\ f_{10}\ f_{11}\ f_6\ f_9\ f_{12})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 13	6 12	7 11	8 9	10 14
1 6	2 3	4 14	5 11	7 10	8 12	9 13
1 7	2 10	3 11	4 5	6 13	8 14	9 12
1 8	2 9	3 6	4 12	5 14	7 13	10 11
1 9	2 7	3 12	4 8	5 13	6 10	11 14
1 10	2 8	3 7	4 11	5 9	6 14	12 13
1 11	2 13	3 5	4 9	6 8	7 14	10 12
1 12	2 14	3 10	4 6	5 7	8 13	9 11
1 13	2 12	3 14	4 10	5 8	6 11	7 9
1 14	2 11	3 9	4 13	5 12	6 7	8 10

Set 11: $|A| = 6$

$A \cong \mathbb{Z}_6$

$A = \langle g1 \rangle$

$g1 = (3\ 7\ 11\ 4\ 12\ 8)(5\ 10\ 14\ 6\ 9\ 13)$

g1 induces

$(f_3\ f_{11}\ f_7\ f_4\ f_{12}\ f_8)(f_5\ f_{10}\ f_{14}\ f_6\ f_9\ f_{13})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 8	3 10	4 13	6 12	7 14	9 11
1 6	2 7	3 14	4 9	5 11	8 13	10 12
1 7	2 14	3 9	4 11	5 13	6 10	8 12
1 8	2 13	3 12	4 10	5 9	6 14	7 11
1 9	2 4	3 5	6 11	7 10	8 14	12 13
1 10	2 3	4 6	5 12	7 13	8 9	11 14
1 11	2 10	3 7	4 12	5 14	6 8	9 13
1 12	2 9	3 11	4 8	5 7	6 13	10 14
1 13	2 12	3 6	4 14	5 8	7 9	10 11
1 14	2 11	3 13	4 5	6 7	8 10	9 12

Set 12: $|A| = 6$

$A \cong \mathbf{Z}_6$

$A = \langle g1 \rangle$

$g1 = (3\ 8\ 13\ 4\ 7\ 14)(5\ 9\ 12\ 6\ 10\ 11)$

g1 induces $(f_3\ \ f_8\ \ f_{13}\ \ f_4\ \ f_7\ \ f_{14})$

$(f_5\ \ f_9\ \ f_{12}\ \ f_6\ \ f_{10}\ \ f_{11})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 8	3 13	4 6	7 11	9 12	10 14
1 6	2 7	3 5	4 14	8 12	9 13	10 11
1 7	2 10	3 12	4 5	6 14	8 13	9 11
1 8	2 9	3 6	4 11	5 13	7 14	10 12
1 9	2 13	3 11	4 8	5 14	6 12	7 10
1 10	2 14	3 7	4 12	5 11	6 13	8 9
1 11	2 3	4 10	5 9	6 7	8 14	12 13
1 12	2 4	3 9	5 8	6 10	7 13	11 14
1 13	2 12	3 14	4 9	5 7	6 11	8 10
1 14	2 11	3 10	4 13	5 12	6 8	7 9

Set 13: $|A| = 156$ (GK_{14})

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 9	3 7	4 11	6 13	8 14	10 12
1 6	2 10	3 12	4 8	5 14	7 13	9 11
1 7	2 13	3 11	4 14	5 9	6 12	8 10
1 8	2 14	3 13	4 12	5 11	6 10	7 9
1 9	2 12	3 14	4 10	5 13	6 8	7 11
1 10	2 11	3 9	4 13	5 7	6 14	8 12
1 11	2 8	3 10	4 6	5 12	7 14	9 13
1 12	2 7	3 5	4 9	6 11	8 13	10 14
1 13	2 4	3 6	5 8	7 10	9 12	11 14
1 14	2 3	4 5	6 7	8 9	10 11	12 13

Set 14: $|A| = 12$

$A \cong \mathbf{Z}_{12}$

$A = \langle g1 \rangle$

g1 = (3 5 9 8 14 12 4 6 10 7 13 11)

g1 induces

$(f_3 \; f_5 \; f_9 \; f_8 \; f_{14} \; f_{12} \; f_4 \; f_6 \; f_{10} \; f_7 \; f_{13} \; f_{11})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 9	3 14	4 12	6 13	7 11	8 10
1 6	2 10	3 11	4 13	5 14	7 9	8 12
1 7	2 13	3 10	4 8	5 9	6 12	11 14
1 8	2 14	3 7	4 9	5 11	6 10	12 13
1 9	2 8	3 13	4 6	5 12	7 14	10 11
1 10	2 7	3 5	4 14	6 11	8 13	9 12
1 11	2 3	4 5	6 7	8 14	9 13	10 12
1 12	2 4	3 6	5 8	7 13	9 11	10 14
1 13	2 11	3 12	4 10	5 7	6 14	8 9
1 14	2 12	3 9	4 11	5 13	6 8	7 10

Set 15: $|A| = 12$

$A \cong \mathbb{Z}_{12}$

$A = \langle g1 \rangle$

g1 = (3 5 14 9 7 11 4 6 13 10 8 12)

g1 induces

$(f_3\ f_5\ f_{14}\ f_9\ f_7\ f_{11}\ f_4\ f_6\ f_{13}\ f_{10}\ f_8\ f_{12})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 14	3 9	4 12	6 11	7 13	8 10
1 6	2 13	3 11	4 10	5 12	7 9	8 14
1 7	2 11	3 5	4 9	6 12	8 13	10 14
1 8	2 12	3 10	4 6	5 11	7 14	9 13
1 9	2 7	3 12	4 8	5 13	6 10	11 14
1 10	2 8	3 7	4 11	5 9	6 14	12 13
1 11	2 4	3 13	5 14	6 7	8 9	10 12
1 12	2 3	4 14	5 8	6 13	7 10	9 11
1 13	2 10	3 14	4 5	6 8	7 11	9 12
1 14	2 9	3 6	4 13	5 7	8 12	10 11

Set 16: $|A| = 3$

$A \cong \mathbb{Z}_3$

$A = \langle g1 \rangle$

$g1 = (1\ 5\ 14)(3\ 11\ 13)(4\ 8\ 7)(6\ 12\ 10)$

g1 induces

$(f_2\ f_3\ f_{10})(f_4\ f_9\ f_{12})(f_5\ f_{13}\ f_{14})(f_6\ f_7\ f_{11})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 10	6 12	7 9	8 13	11 14
1 6	2 3	4 12	5 13	7 10	8 14	9 11
1 7	2 11	3 14	4 6	5 12	8 10	9 13
1 8	2 9	3 12	4 14	5 7	6 13	10 11
1 9	2 12	3 13	4 10	5 8	6 14	7 11
1 10	2 14	3 6	4 8	5 11	7 13	9 12
1 11	2 13	3 9	4 5	6 7	8 12	10 14
1 12	2 10	3 11	4 13	5 9	6 8	7 14
1 13	2 8	3 7	4 9	5 14	6 11	10 12
1 14	2 7	3 5	4 11	6 10	8 9	12 13

Set 17: $|A| = 3$

$A \cong \mathbb{Z}_3$

$A = \langle g1 \rangle$

g1 = (1 8 3)(2 9 14)(4 11 5)(7 13 10)(6)(12)

g1 induces

$(f_2\ f_{11}\ f_{10})(f_3\ f_8\ f_4)(f_5\ f_{14}\ f_9)(f_6\ f_{12}\ f_{13})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 13	9 12	11 14
1 5	2 4	3 9	6 13	7 12	8 14	10 11
1 6	2 14	3 12	4 5	7 11	8 10	9 13
1 7	2 11	3 10	4 14	5 9	6 12	8 13
1 8	2 12	3 5	4 9	6 14	7 10	11 13
1 9	2 8	3 11	4 13	5 14	6 7	10 12
1 10	2 13	3 14	4 12	5 8	6 11	7 9
1 11	2 10	3 13	4 6	5 12	7 14	8 9
1 12	2 9	3 7	4 11	5 13	6 8	10 14
1 13	2 7	3 6	4 10	5 11	8 12	9 14
1 14	2 3	4 8	5 7	6 10	9 11	12 13

Set 18: $|A| = 3$

$A \cong \mathbb{Z}_3$

$A = \langle g1 \rangle$

g1 = (1 5 9)(2 7 14)(4 6 8)(10 13 11)

g1 induces

$(f_2\ f_{13}\ f_4)(f_3\ f_8\ f_{14})(f_5\ f_{10}\ f_9)(f_7\ f_{11}\ f_{12})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 11	7 10	9 14	12 13
1 5	2 4	3 13	6 12	7 14	8 10	9 11
1 6	2 13	3 12	4 9	5 8	7 11	10 14
1 7	2 10	3 14	4 11	5 12	6 8	9 13
1 8	2 12	3 5	4 10	6 14	7 9	11 13
1 9	2 7	3 10	4 12	5 13	6 11	8 14
1 10	2 14	3 11	4 13	5 9	6 7	8 12
1 11	2 3	4 8	5 14	6 10	7 13	9 12
1 12	2 9	3 7	4 6	5 10	8 13	11 14
1 13	2 11	3 6	4 14	5 7	8 9	10 12
1 14	2 8	3 9	4 5	6 13	7 12	10 11

Set 19: $|A| = 3$

$A \cong \mathbf{Z}_3$

$A = \langle g1 \rangle$

g1 = (1 4 7)(2 11 5)(3 12 10)(6 8 14)

g1 induces

$(f_2\ f_{14}\ f_{10})(f_3\ f_7\ f_4)(f_5\ f_9\ f_{12})(f_6\ f_{11}\ f_{13})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 11	7 10	9 14	12 13
1 5	2 13	3 14	4 10	6 11	7 9	8 12
1 6	2 7	3 10	4 13	5 12	8 14	9 11
1 7	2 11	3 13	4 12	5 14	6 10	8 9
1 8	2 12	3 5	4 14	6 7	9 13	10 11
1 9	2 4	3 7	5 8	6 12	10 14	11 13
1 10	2 3	4 6	5 7	8 13	9 12	11 14
1 11	2 10	3 12	4 8	5 9	6 14	7 13
1 12	2 14	3 6	4 9	5 13	7 11	8 10
1 13	2 9	3 11	4 5	6 8	7 14	10 12
1 14	2 8	3 9	4 11	5 10	6 13	7 12

Set 20: $|A| = 3$

$A \cong \mathbb{Z}_3$

$A = \langle g1 \rangle$

g1 = (1 8 11)(2 9 10)(3 5 4)(6 12 14)

g1 induces

$(f_2\ f_{14}\ f_7)(f_3\ f_{11}\ f_8)(f_5\ f_{13}\ f_6)(f_9\ f_{12}\ f_{10})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 11	7 13	9 12	10 14
1 5	2 7	3 6	4 10	8 12	9 14	11 13
1 6	2 12	3 11	4 14	5 9	7 10	8 13
1 7	2 9	3 14	4 5	6 8	10 11	12 13
1 8	2 14	3 10	4 11	5 7	6 12	9 13
1 9	2 8	3 12	4 13	5 10	6 11	7 14
1 10	2 11	3 9	4 6	5 13	7 12	8 14
1 11	2 13	3 7	4 9	5 8	6 14	10 12
1 12	2 4	3 13	5 14	6 7	8 10	9 11
1 13	2 3	4 8	5 12	6 10	7 9	11 14
1 14	2 10	3 5	4 12	6 13	7 11	8 9

Set 21: $|A| = 2$

$A \cong \mathbf{Z}_2$

$A = \langle g1 \rangle$

g1 = (1 4)(2 5)(3 7)(6 10)(8 12)(9 13)(11 14)

g1 induces $(f_2\ f_7)(f_6\ f_{12})(f_9\ f_{14})$

1 2	3 4	5 6	7 8	9 10	11 12	13 14
1 3	2 5	4 7	6 9	8 11	10 13	12 14
1 4	2 6	3 8	5 10	7 12	9 14	11 13
1 5	2 4	3 9	6 11	7 13	8 12	10 14
1 6	2 14	3 5	4 8	7 9	10 11	12 13
1 7	2 10	3 12	4 5	6 13	8 14	9 11
1 8	2 11	3 6	4 12	5 14	7 10	9 13
1 9	2 13	3 10	4 11	5 12	6 8	7 14
1 10	2 3	4 6	5 7	8 13	9 12	11 14
1 11	2 9	3 7	4 14	5 13	6 12	8 10
1 12	2 7	3 13	4 10	5 11	6 14	8 9
1 13	2 12	3 14	4 9	5 8	6 10	7 11
1 14	2 8	3 11	4 13	5 9	6 7	10 12

APPENDIX 3

Frequency distribution of non-isomorphic set of two mutually orthogonal OFs 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.

APPENDIX 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

APPENDIX 5

$H_4(8, 10)$

(F, G), (F, H), (G, H): see Appendix 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

APPENDIX 6

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

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

Set 1 $A = < I >$.

Set 2 $A = < g > \cong \mathbb{Z}_8$, where $g = (3\ 5\ 8\ 10\ 4\ 6\ 7\ 9)$.

g interchanges G and H.

Set 3 $A = < g > \cong \mathbb{Z}_6$, where $g = (5\ 6)(3\ 8\ 10\ 4\ 7\ 9)$.

g maps F into G, G into H, and H into F.

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

$A = < g_1, g_2 >, |A| = 24,$

and $g_1 = (3\ 4)(5\ 10\ 8\ 6\ 9\ 7)$,

$g_2 = (5\ 6)(3\ 8\ 10\ 4\ 7\ 9)$.

g_1 maps H into G, G into I, and I into H,

g_2 maps F into G, G into H, and H into F.

APPENDIX 7

Three skew H(8, 10) designs

a = 5, b = 6

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

a = 5, b = 6

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

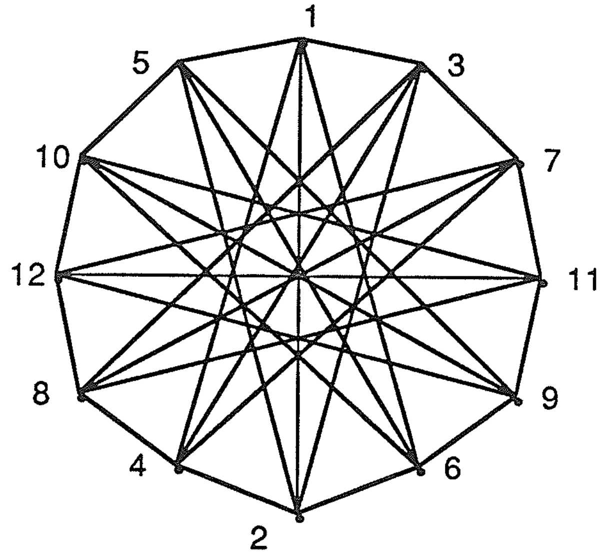
a = 7, b = 8

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

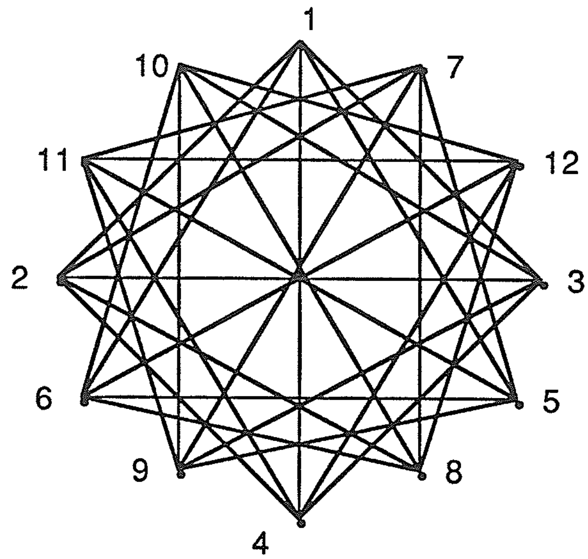
APPENDIX 8

5-regular graphs on 12 vertices having transitive automorphism groups

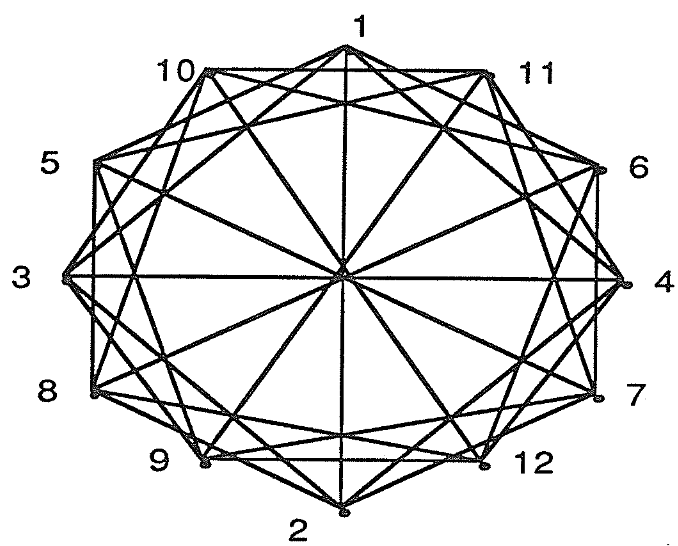
graph no. 1 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 5, 6; 3 - 4, 7, 8; 4 - 7, 8; 5 - 6, 9, 10; 6 - 9, 10; 7 - 8, 11, 12; 8 - 11, 12; 9 - 10, 11, 12; 10 - 11, 12; 11 - 12.



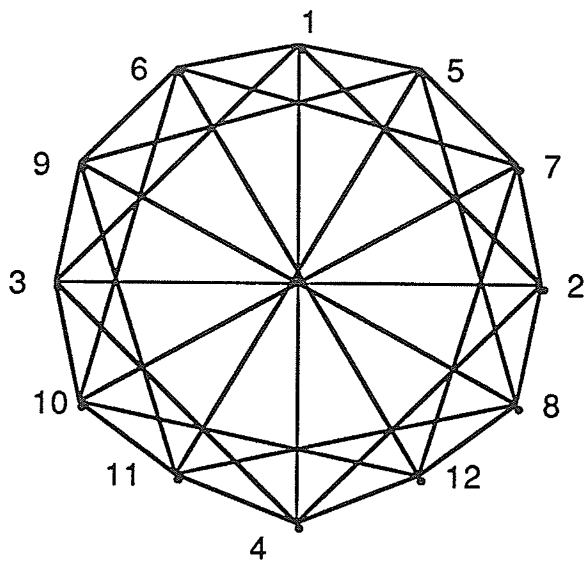
graph no. 2 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 4, 9, 10; 4 - 11, 12; 5 - 6, 7, 9, 11; 6 - 8, 10, 12; 7 - 8, 9, 11; 8 - 10, 12; 9 - 10, 11; 10 - 12; 11 - 12.



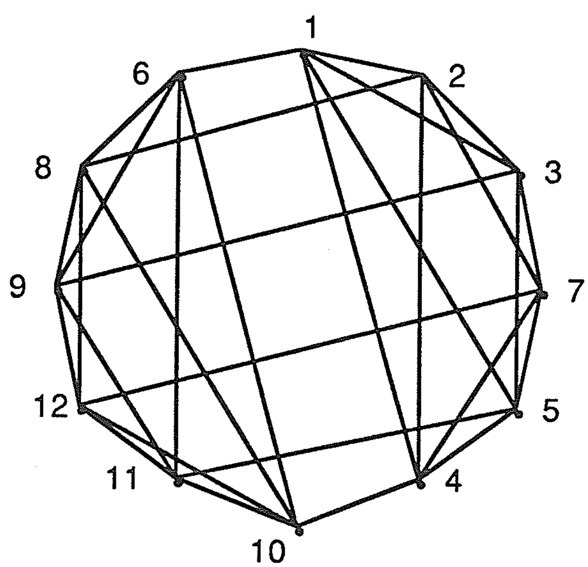
graph no. 3 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 4, 9, 10; 4 - 11, 12; 5 - 7, 8, 9, 11; 6 - 7, 8, 10, 12; 7 - 9, 11; 8 - 10, 12; 9 - 11, 12; 10 - 11, 12.



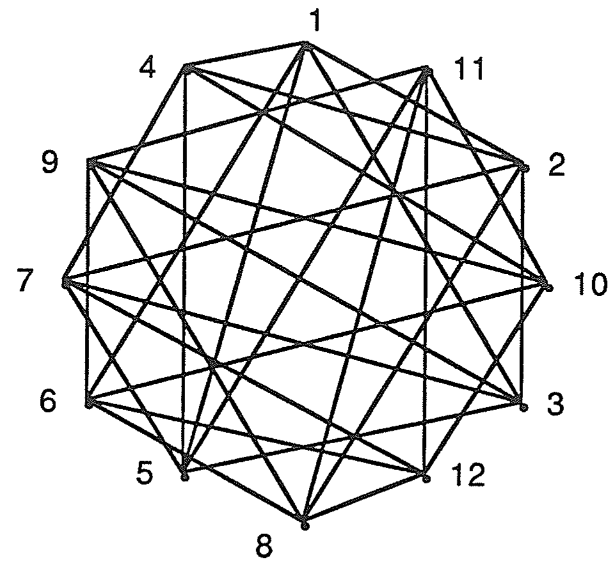
graph no. 4 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 4, 9, 10; 4 - 11, 12; 5 - 7, 8, 9, 11; 6 - 7, 9, 10, 12; 7 - 10, 12; 8 - 9, 11, 12; 9 - 11; 10 - 11, 12.



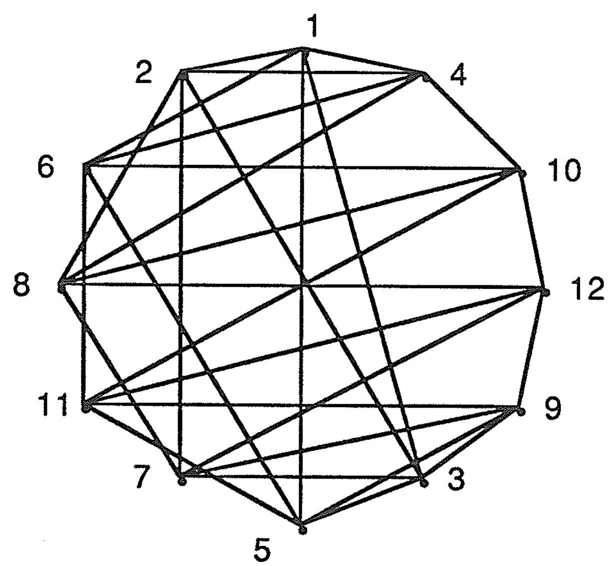
graph no. 5 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 5, 7, 9; 4 - 5, 7, 10; 5 - 7, 11; 6 - 8, 9, 10, 11; 7 - 12; 8 - 9, 10, 12; 9 - 11, 12; 10 - 11, 12; 11 - 12.



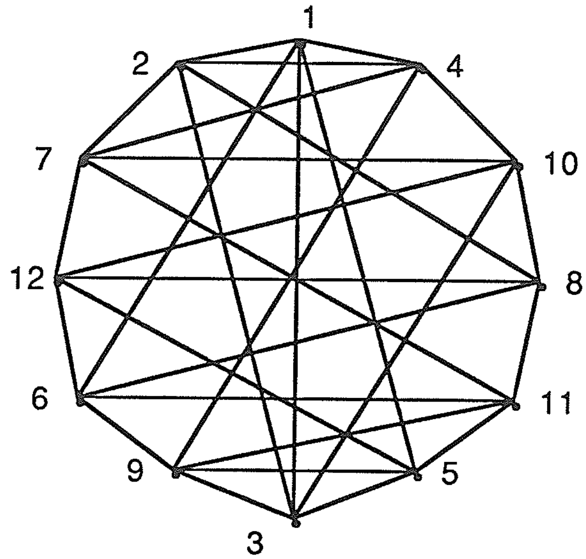
graph no. 6 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 5, 7, 9; 4 - 5, 7, 10; 5 - 7, 11; 6 - 8, 9, 10, 12; 7 - 12; 8 - 9, 11, 12; 9 - 10, 11; 10 - 11, 12; 11 - 12.



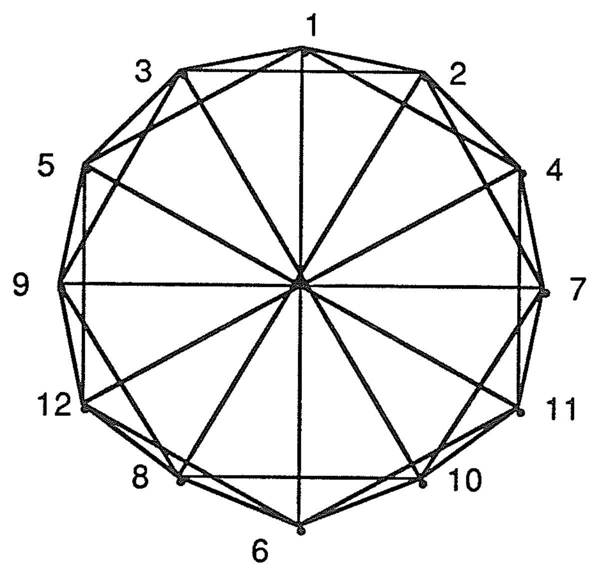
graph no. 7 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 5, 7, 9; 4 - 6, 8, 10; 5 - 6, 9, 11; 6 - 10, 11; 7 - 8, 9, 12; 8 - 10, 12; 9 - 11, 12; 10 - 11, 12; 11 - 12.



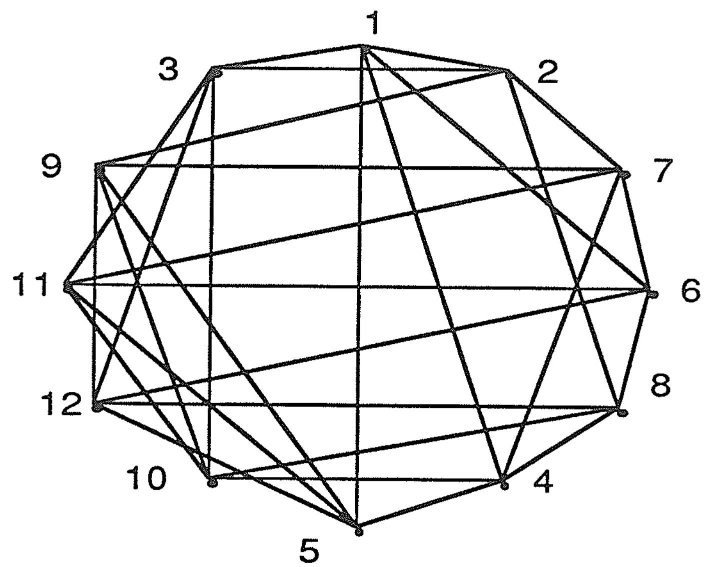
graph no. 8 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 5, 9, 10; 4 - 7, 9, 10; 5 - 9, 11, 12; 6 - 8, 9, 11, 12; 7 - 10, 11, 12; 8 - 10, 11, 12; 9 - 11; 10 - 12.



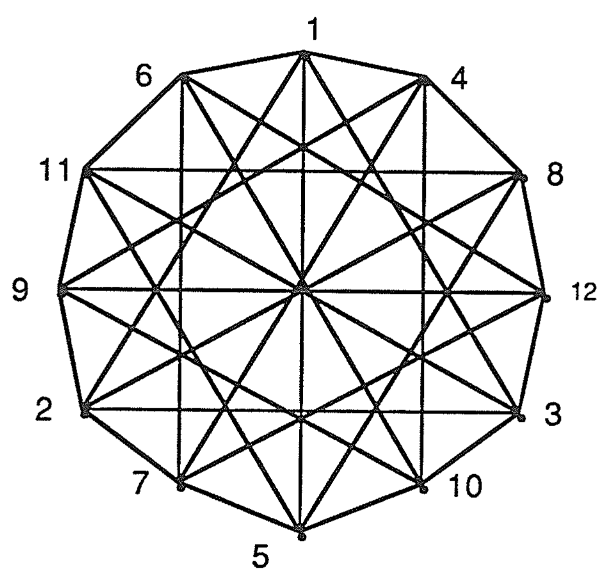
graph no. 9 1 - 2, 3, 4, 5, 6; 2 - 3, 4, 7, 8; 3 - 5, 9, 10; 4 - 7, 11, 12; 5 - 9, 11, 12; 6 - 8, 10, 11, 12; 7 - 9, 10, 11; 8 - 9, 10, 12; 9 - 12; 10 - 11.



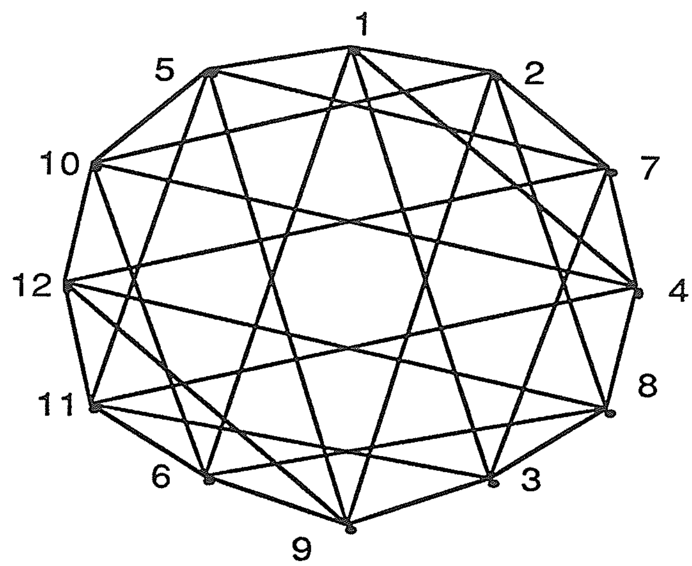
graph no. 10 1 - 2, 3, 4, 5, 6; 2 - 3, 7, 8, 9; 3 - 10, 11, 12; 4 -5, 7, 8, 10; 5 - 9, 11, 12; 6 - 7, 8, 11, 12; 7 - 9, 11; 8 - 10, 12; 9 - 10, 12; 10 - 11.



graph no. 11 1 - 2, 3, 4, 5, 6; 2 - 3, 7, 8, 9; 3 - 10, 11, 12; 4 -7, 8, 9, 10; 5 - 7, 8, 10, 11; 6 - 7, 10, 11, 12; 7 - 12; 8 - 11, 12; 9 - 10, 11, 12.



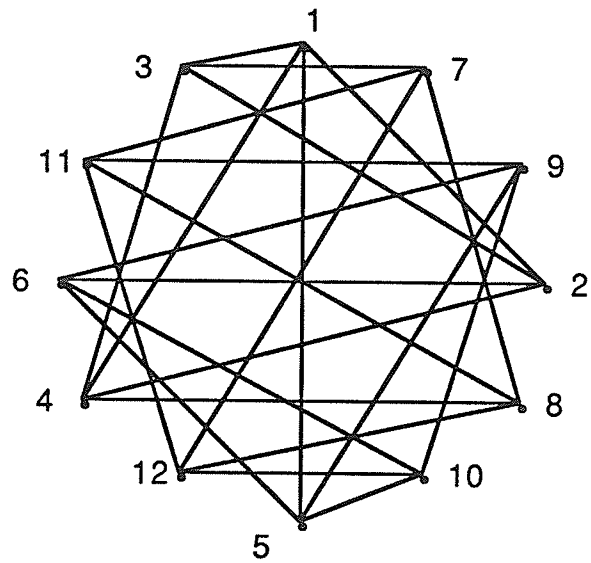
graph no. 12 1 - 2, 3, 4, 5, 6; 2 - 7, 8, 9, 10; 3 - 7, 8, 9, 11; 4 -7, 8, 10, 11; 5 - 7, 9, 10, 11; 6 - 8, 9, 10, 11; 7 - 12; 8 - 12; 9 - 12; 10 - 12; 11 - 12.



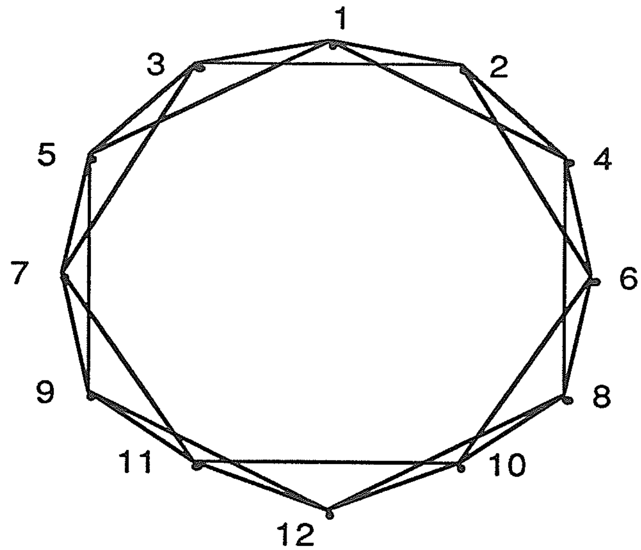
APPENDIX 9

4-regular graphs on 12 vertices having transitive automorphism groups

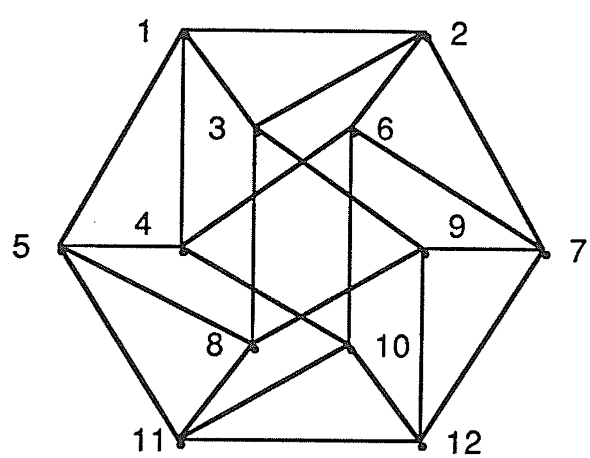
graph no. 1 1 - 2, 3, 4, 5; 2 - 3, 4, 6; 3 - 4, 7; 4 - 8; 5 - 6, 9, 10; 6 - 9, 10; 7 - 8, 11, 12; 8 - 11, 12; 9 - 10, 11; 10 - 12; 11 - 12.



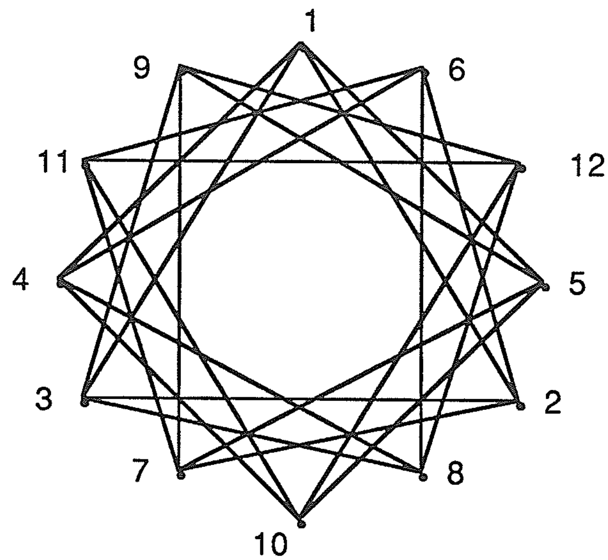
graph no. 2 1 - 2, 3, 4, 5; 2 - 3, 4, 6; 3 - 5, 7; 4 - 6, 8; 5 - 7, 9; 6 - 8, 10; 7 - 9, 11; 8 - 10, 12; 9 - 11, 12; 10 - 11, 12; 11 - 12.



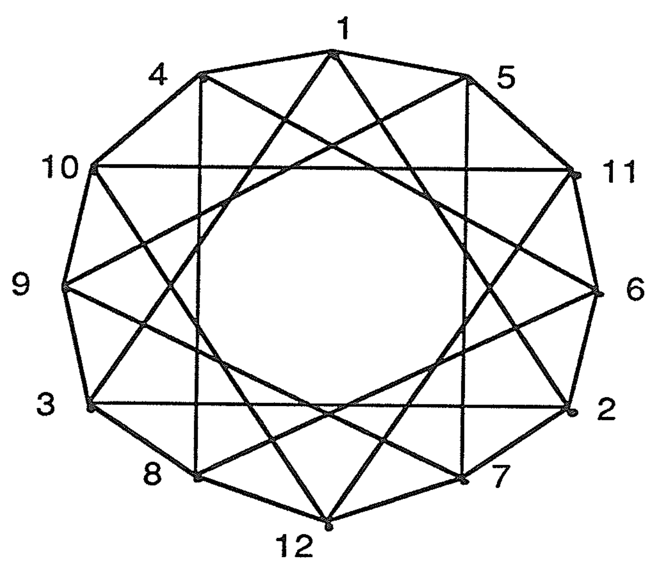
graph no. 3 1 - 2, 3, 4, 5; 2 - 3, 6, 7; 3 - 8, 9; 4 - 5, 6, 10; 5 - 8, 11; 6 - 7, 10; 7 - 9, 12; 8 - 9, 11; 9 - 12; 10 - 11, 12; 11 - 12.



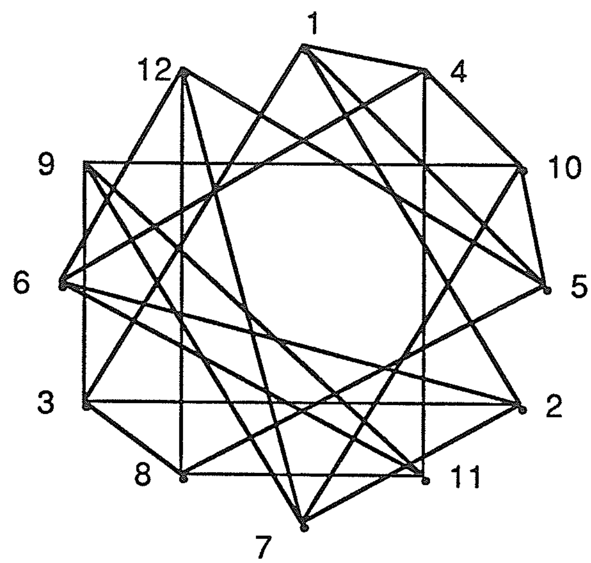
graph no. 4 1 - 2, 3, 4, 5; 2 - 3, 6, 7; 3 - 8, 9; 4 - 6, 8, 10; 5 - 7, 9, 10; 6 - 8, 11; 7 - 9, 11; 8 - 12; 9 - 12; 10 - 11, 12; 11 - 12.



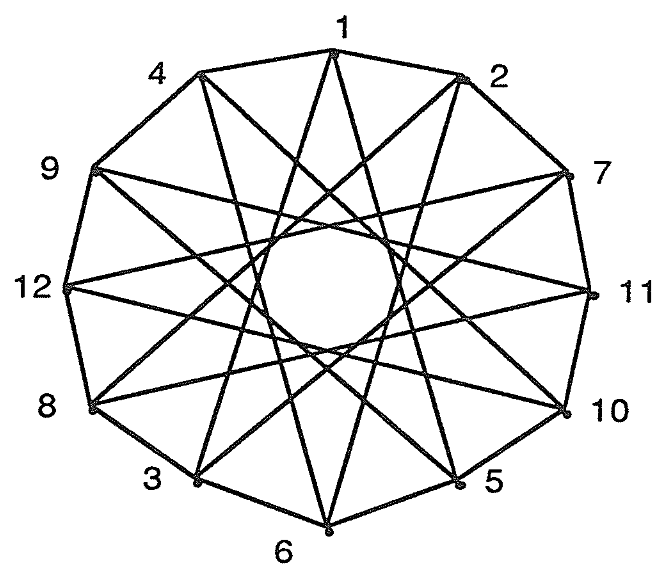
graph no. 5 1 - 2, 3, 4, 5; 2 - 3, 6, 7; 3 - 8, 9; 4 - 6, 8, 10; 5 - 7, 9, 11; 6 - 8, 11; 7 - 9, 12; 8 - 12; 9 - 10; 10 - 11, 12; 11 - 12.



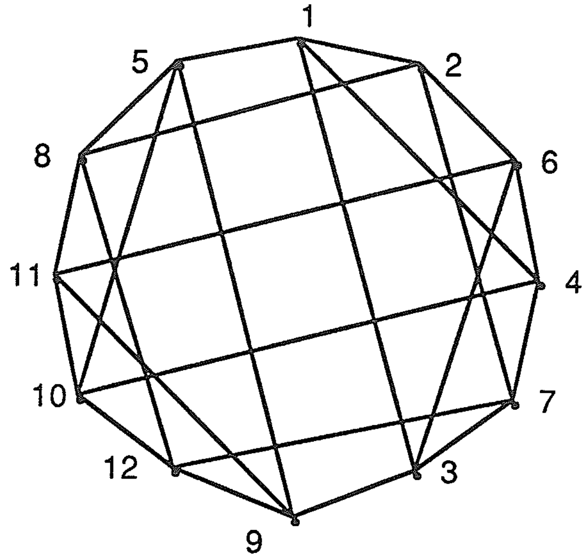
graph no. 6 1 - 2, 3, 4, 5; 2 - 3, 6, 7; 3 - 8, 9; 4 - 6, 10, 11; 5 - 8, 10, 12; 6 - 11, 12; 7 - 9, 10, 12; 8 - 11, 12; 9 - 10, 11.



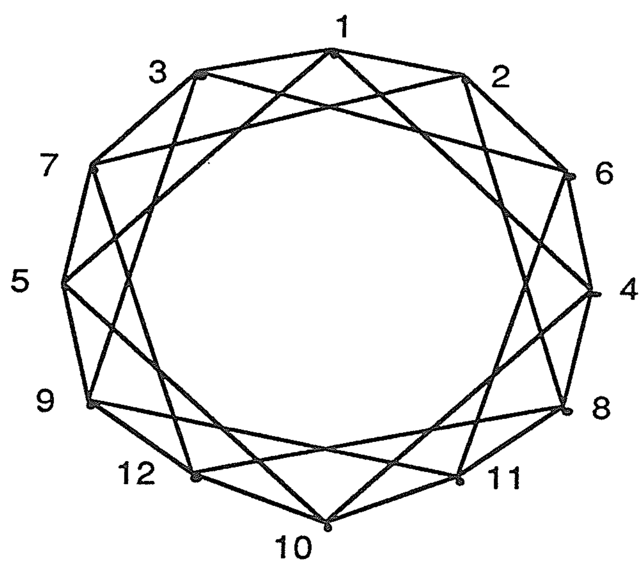
graph no. 7 1 - 2, 3, 4, 5; 2 - 6, 7, 8; 3 - 6, 7, 8; 4 - 6, 9, 10; 5 - 6, 9, 10; 7 - 11, 12; 8 - 11, 12; 9 - 11, 12; 10 - 11, 12.



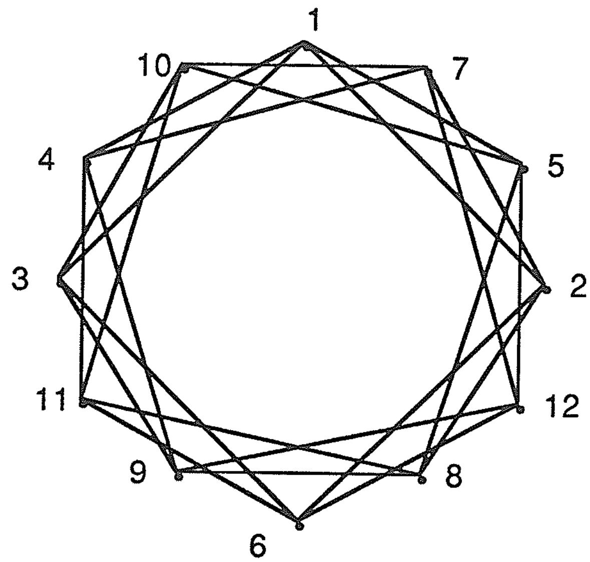
graph no. 8 1 - 2, 3, 4, 5; 2 - 6, 7, 8; 3 - 6, 7, 9; 4 - 6, 7, 10; 5 - 8, 9, 10; 6 - 11; 7 - 12; 8 - 11, 12; 9 - 11, 12; 10 - 11, 12.



graph no. 9 1 - 2, 3, 4, 5; 2 - 6, 7, 8; 3 - 6, 7, 9; 4 - 6, 8, 10; 5 - 7, 9, 10; 6 - 11; 7 - 12; 8 - 11, 12; 9 - 11, 12; 10 - 11, 12.



graph no. 10 1 - 2, 3, 4, 5; 2 - 6, 7, 8; 3 - 6, 9, 10; 4 - 7, 9, 11; 5 - 8, 10, 12; 6 - 11, 12; 7 - 10, 12; 8 - 9, 11; 9 - 12; 10 - 11.



APPENDIX 10

A Howell cube $H_3(7, 12)$

1 2	6 11	5 10	7 12	3 4	8 9	
5 11	1 3	6 7	2 9	8 12		4 10
6 8	2 5	1 4		7 11	3 10	9 12
10 12	7 9	2 8	1 5		4 6	3 11
	8 10	3 9	4 11	1 6	5 12	2 7
4 9		11 12	3 8	2 10	1 7	5 6
3 7	4 12		6 10	5 9	2 11	1 8

1 2		6 11	3 4	8 9	5 10	7 12
8 12	1 3		6 7	4 10	2 9	5 11
3 10	7 11	1 4	9 12	2 5	6 8	
7 9	2 8	10 12	1 5		3 11	4 6
4 11	5 12	2 7	8 10	1 6		3 9
5 6	4 9	3 8		11 12	1 7	2 10
	6 10	5 9	2 11	3 7	4 12	1 8

1 2	4 9	10 12		3 7	6 8	5 11
7 9	1 3	6 11	8 10	2 5	4 12	
	2 8	1 4	6 7	11 12	5 10	3 9
4 11	6 10	3 8	1 5		2 9	7 12
8 12	7 11	5 9	3 4	1 6		2 10
3 10	5 12		2 11	8 9	1 7	4 6
5 6		2 7	9 12	4 10	3 11	1 8

APPENDIX 11

Three Howell designs H*(7, 12)

1 2	7 11		6 10	3 4	8 12	5 9
7 12	1 3	8 10	4 9	2 5	6 11	
8 9		1 4	11 12	7 10	3 5	2 6
4 11	2 8	6 12	1 5		9 10	3 7
	5 12	7 9	3 8	1 6	2 4	10 11
5 10	6 9	2 3		8 11	1 7	4 12
3 6	4 10	5 11	2 7	9 12		1 8

1 2		6 10	3 4	5 9	8 12	7 11
	1 3	8 11	7 12	4 10	6 9	2 5
5 12	7 9	1 4	10 11	2 8		3 6
3 8	6 12	2 7	1 5		4 11	9 10
7 10	5 11		8 9	1 6	2 3	4 12
4 9	8 10	3 5	2 6	11 12	1 7	
6 11	2 4	9 12		3 7	5 10	1 8

1 2		8 12	3 4	5 9	6 10	7 11
	1 3	7 9	6 12	4 10	8 11	2 5
6 9	8 10	1 4	2 7	11 12	3 5	
3 8	7 12	6 11	1 5		2 4	9 10
7 10	5 11	2 3	8 9	1 6		4 12
5 12	4 9		10 11	2 8	1 7	3 6
4 11	2 6	5 10		3 7	9 12	1 8

APPENDIX 12

Two sets of almost disjoint Howell designs H(7, 14)

Set 1: $\{D_1, D_2\}$.

D_1

a	3	$\underline{a}$	$\underline{3}$	$\underline{2}$	$\underline{4}$	2	4	$\underline{1}$	$\underline{5}$	1	5	6	$\underline{6}$
$\underline{a}$	2	a	$\underline{2}$	$\underline{1}$	$\underline{3}$	1	3	$\underline{4}$	$\underline{6}$	4	6	5	$\underline{5}$
$\underline{1}$	$\underline{2}$	1	2	a	$\underline{5}$	$\underline{a}$	5	3	6	$\underline{3}$	$\underline{6}$	4	$\underline{4}$
$\underline{3}$	$\underline{4}$	3	4	$\underline{a}$	$\underline{6}$	a	6	2	5	$\underline{2}$	$\underline{5}$	1	$\underline{1}$
4	5	$\underline{4}$	$\underline{5}$	2	6	$\underline{2}$	$\underline{6}$	a	1	$\underline{a}$	$\underline{1}$	3	$\underline{3}$
1	6	$\underline{1}$	$\underline{6}$	3	5	$\underline{3}$	$\underline{5}$	$\underline{a}$	4	a	$\underline{4}$	2	$\underline{2}$
$\underline{5}$	$\underline{6}$	5	6	1	4	$\underline{1}$	$\underline{4}$	$\underline{2}$	$\underline{3}$	2	3	a	$\underline{a}$

D_2

$\underline{a}$	$\underline{4}$	3	$\underline{1}$	5	$\underline{3}$	1	$\underline{5}$	a	2	4	$\underline{2}$	6	$\underline{6}$
1	$\underline{2}$	2	$\underline{4}$	4	$\underline{6}$	$\underline{a}$	3	6	$\underline{1}$	a	$\underline{3}$	5	$\underline{5}$
3	$\underline{6}$	6	$\underline{5}$	a	$\underline{1}$	2	$\underline{3}$	5	$\underline{2}$	$\underline{a}$	1	4	$\underline{4}$
6	$\underline{3}$	a	4	3	$\underline{2}$	5	$\underline{4}$	$\underline{a}$	$\underline{5}$	2	$\underline{6}$	1	$\underline{1}$
a	5	$\underline{a}$	$\underline{2}$	2	$\underline{5}$	4	$\underline{1}$	1	$\underline{6}$	6	$\underline{4}$	3	$\underline{3}$
4	$\underline{5}$	1	$\underline{3}$	$\underline{a}$	6	a	$\underline{6}$	3	$\underline{4}$	5	$\underline{1}$	2	$\underline{2}$
2	$\underline{1}$	5	$\underline{6}$	1	$\underline{4}$	6	$\underline{2}$	4	$\underline{3}$	$\underline{5}$	$\underline{3}$	a	$\underline{a}$

Set 2: $\{D_1, D_3\}$.

D_3

a	4	$\underline{a}$	$\underline{4}$	2	$\underline{3}$	1	$\underline{5}$	5	$\underline{2}$	3	$\underline{1}$	6	$\underline{6}$
6	$\underline{3}$	3	$\underline{2}$	a	$\underline{6}$	4	$\underline{1}$	2	$\underline{4}$	$\underline{a}$	1	5	$\underline{5}$
1	$\underline{2}$	2	$\underline{5}$	5	$\underline{1}$	3	$\underline{6}$	$\underline{a}$	6	a	$\underline{3}$	4	$\underline{4}$
3	$\underline{4}$	a	5	6	$\underline{5}$	$\underline{a}$	$\underline{2}$	4	$\underline{3}$	2	$\underline{6}$	1	$\underline{1}$
$\underline{a}$	$\underline{5}$	6	$\underline{1}$	4	$\underline{2}$	a	2	1	$\underline{6}$	5	$\underline{4}$	3	$\underline{3}$
5	$\underline{6}$	1	$\underline{3}$	$\underline{a}$	3	6	$\underline{4}$	a	$\underline{1}$	4	$\underline{5}$	2	$\underline{2}$
2	$\underline{1}$	4	$\underline{6}$	1	$\underline{4}$	5	$\underline{3}$	3	$\underline{5}$	6	$\underline{2}$	a	$\underline{a}$

APPENDIX 13

Non-isomorphic almost disjoint Howell designs H(5, 10)

Set 1: $\{D_1, D_2\}$. $f = \{(1\ 9), (2\ 10), (3\ 7), (4\ 8), (5\ 6)\}$.

Underlying graph of D_1 is graph no. 2;

Underlying graph of D_2 is graph no. 17.

D_1	1 2	3 4	5 7	6 9	8 10
	3 7	1 9	2 10	4 8	5 6
	4 6	2 5	8 9	7 10	1 3
	5 8	6 10	1 4	2 3	7 9
	9 10	7 8	3 6	1 5	2 4

D_2	1 6	5 10	4 9	2 7	3 8
	4 10	1 7	3 5	6 8	2 9
	5 9	2 6	1 8	3 10	4 7
	3 7	4 8	2 10	1 9	5 6
	2 8	3 9	6 7	4 5	1 10

Set 2: $\{D_1, D_2\}$. $f = \{(1\ 10), (2\ 8), (3\ 9), (4\ 6), (5\ 7)\}$.

Underlying graph of D_1 is graph no. 17;

Underlying graph of D_2 is graph no. 50.

D_1	1 2	3 5	4 6	7 9	8 10
	9 10	2 6	5 7	4 8	1 3
	3 7	1 4	2 8	6 10	5 9
	6 8	7 10	3 9	1 5	2 4
	4 5	8 9	1 10	2 3	6 7

D_2	1 6	4 10	2 5	7 8	3 9
	5 10	1 7	6 9	3 4	2 8
	4 9	3 6	1 8	2 10	5 7
	2 7	5 8	3 10	1 9	4 6
	3 8	2 9	4 7	5 6	1 10

REFERENCES

1 B. A. Anderson, Finite topologies and Hamiltonian paths, *Journal of Comb. Theory* B 14 (1973), 87-93.

2 B. A. Anderson, "Some perfect 1-factorizations", Proc. of 7th Southeastern Conference on Combinatorics, Graph Theory and Computing, Utilitas Math., Winnipeg (1976), 79-91.

3 B. A. Anderson, Symmetry Groups of Some Perfect One-factorizations of Complete Graphs, *Discrete Math.* 18 (1977), 227-234.

4 B. A. Anderson, Sequencings of Dicyclic Groups, *Ars Combinatoria* 23 (1987), 131-142.

5 B. A. Anderson, M. M. Barge and D. Morse, A recursive construction of asymmetric one-factorizations, *Aequationes Math.* 15 (1977), 201-211.

6 B. A. Anderson, P. J. Schellenberg and D. R. Stinson, The existence of Howell designs of even side, *Journal of Comb. Theory* A 36 (1984), 23-55.

7 D. S. Archdeacon, J. H. Dinitz and W. D. Wallis, Sets of orthogonal 1-factorizations of K_{10}, *Congressus Numerantium* 43 (1984), 45-79.

8 A. M. Barayev and I. A. Faradzev, Postroenie i issledovanie na EVM odnorodnykh i odnorodnykh dvudol'nykh grafov, *Algoritmiceskie issledovania v kombinatorike*, Moscow, Nauka (1978), 25-60.

9 L. R. Beaman, "Variability of Room squares", Ph. D. Thesis, University of Newcastle (1979).

10 R. C. Bose, E. T. Parker and S. S. Shrikhande, Further results on the construction of mutually orthogonal Latin squares and the falsity of Euler's conjecture, *Canad. Journal Math.* 12 (1960), 189-203.

11 E. F. Brickell, A few results in message authentication, *Congressus Numerantium* 43 (1984), 141-154.

12 J. W. Brown, Enumeration of Latin squares with application to order 8, *Journal of Comb. Theory* B 5 (1968), 177-184.

13 P. J. Cameron, Minimal edge-colourings of complete graphs, *Journal London Math. Soc.* (2) 11 (1975), 337-346.

14 P. J. Cameron, "Parallelisms of complete designs", London Math. Soc. Lecture Note Ser. 23, Cambridge University Press, Cambridge 1976.

15 A. G. Chetwynd and A. J. W. Hilton, Regular graphs of high degree are 1-factorizable, *Proc. London Math. Soc.* (3), 50 (1985), 193-206.

16 C. J. Colbourn and M. J. Colbourn, Combinatorial isomorphism problems involving 1-factorizations, *Ars Combinatoria* 9 (1980), 191-200.

17 L. E. Dickson and F. H. Safford, Solution to problem 8 (group theory), *Amer. Math. Monthly* 13 (1906), 150-151.

18 J. H. Dinitz, Pairwise orthogonal symmetric Latin squares, *Congressus Numerantium* 32 (1981), 261-265.

19 J. H. Dinitz and D. R. Stinson, The spectrum of Room cubes, *Europ. Journal of Comb.* 2 (1981), 221-230.

20 J. H. Dinitz and D. R. Stinson, A fast algorithm for finding strong starters, *SIAM Journal of Alg. and Discrete Math.* 2 (1981), 50-56.

21 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.

22 E. N. Gelling, "On 1-factorizations of the complete graph and the relationship to round robin schedules", M.A. Thesis, University of Victoria, 1973.

23 P. B. Gibbons, "Computing techniques for the construction and analysis of block designs", Ph. D. thesis, University of Toronto, 1976.

24 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.

25 A. Hartman and A. Rosa, Cyclic one-factorizatons of the complete graph, *Europ. Journal of Comb.* (1985), 45-48.

26 I. Holyer, NP-completeness of edge-coloring, *SIAM Journal Computing* 10 (1981), 718-720.

27 J. D. Horton, Room designs and one-factorizations, *Aequationes Math.* 22 (1981), 56-63.

28 S. H. Y. Hung and N. S. Mendelsohn, On Howell designs, *Journal of Comb. Theory* A 16 (1974), 174-198.

29 E. C. Ihrig, Symmetry groups related to the construction of perfect one-factorizations of K_{2n}, *Journal of Comb. Theory* B 40 (1986), 121-151.

30 E. C. Ihrig, The structure of the symmetry groups of perfect one-factorizations of K_{2n}, to appear.

31 E. C. Ihrig, Cyclic perfect one-factorizations of K_{2n}, preprint.

32 E. C. Ihrig, E. Seah and D. R. Stinson, A perfect one-factorization of K_{50}, *Journal Comb. Math. and Comb. Comput.*, 1 (1987), 217-219.

33 W. L. Kocay, D. R. Stinson and S. A. Vanstone, On strong starters in cyclic groups, *Discrete Math.* 56 (1985), 45-60.

34 N. P. Korovina, On the construction of closed triples of pair groups of order 12 (in Russian), *Kombinatornyi Anal.* 2 (1972), 42-45.

35 A. Kotzig, "Hamilton graphs and Hamilton circuits", Theory of graphs and its application, Proc. Sympos. Smolenice (1963), Nake. CSAV, Praha (1964), 63-82.

36 E. R. Lamken and S. A. Vanstone, Complementary Howell designs of side 2n and order 2n+2, *Congressus Numerantium* 41 (1984), 85-113.

37 E. R. Lamken and S. A. Vanstone, Partitioned balanced tournament designs of side 4n+1, *Ars Combinatoria* 20 (1985), 29-44.

38 E.R. Lamken and S. A. Vanstone, Designs with mutually orthogonal resolutions, *Europ. Journal of Comb.* 7 (1986), 249-257.

39 E. R. Lamken and S. A. Vanstone, The existence of partitioned balanced tournament designs of side 4n+3, *Annals of Discrete Math.*, to appear.

40 E. R. Lamken and S. A. Vanstone, The existence of partitioned balanced tournament designs, *Annals of Discrete Math.*, to appear.

41 C. C. Lindner, E. Mendelsohn and A. Rosa, On the number of 1-factorizations of the complete graph, *Journal of Comb. Theory* B 20 (1976), 265-282.

42 E. Lucas, Sixième récréation: Les jeux de demoiselles, *Récréations Mathématiques*, Vol. 2, Gauthier-Villars, Paris (1883), 161-197.

43 E. Mendelsohn and A. Rosa, One-factorizations of the complete graph - a survey, *Journal Graph Theory* 9 (1985), 43-65.

44 R. C. Mullin and W. D. Wallis, The existence of Room squares, *Aequationes Math.* 13 (1975), 1-7.

45 C. Papadimtriou and K. Steiglitz, Chapter 19 in "Combinatorial optimization: algorithm and complexity", Prentice Hall, 1982.

46 K. T. Phelps and S. A. Vanstone, Isomorphism of strong starters in cyclic groups, preprint.

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

48 M. Reiss, Über eine Steinersche combinatorische Aufgabe, welche im 45sten Bande dieses Journals, Seite 181, gestellt worden ist, *J. fur die reine angew. Math.* 56 (1859), 326-344.

49 T. G. Room, A new type of magic square, *Math. Gaz.* 39 (1955), 307.

50 A. Rosa, Room squares generalized, *Annals of Discrete Math.* 8 (1980), 43-57.

51 A. Rosa and D. R. Stinson, One-factorizations of regular graphs and Howell designs of small order, *Utilitas Math.* 29 (1986), 99-124.

52 P. J. Schellenberg, D. R. Stinson, S. A. Vanstone and J. W. Yates, The existence of Howell designs of side n+1 and order 2n, *Combinatorica* 1 (1981), 289-301.

53 P. J. Schellenberg and S. A. Vanstone, The existence of Howell designs of side 2n and order 2n+2, *Congressus Numerantium* 29 (1980), 879-887.

54 E. Seah and D. R. Stinson, An enumeration of non-isomorphic one-factorizations and Howell designs for the graph K_{10} minus a one-factor, *Ars Combinatoria* 21 (1986), 145-161.

55 E. Seah and D. R. Stinson, Some perfect one-factorizations of K_{14}, *Annals of Discrete Math.*, to appear.

56 E. Seah and D. R. Stinson, An assortment of new Howell designs, *Utilitas Math.*, to appear.

57 E. Seah and D. R. Stinson, A perfect one-factorization for K_{36}, *Discrete Math.*, to appear.

58 E. Seah and D. R. Stinson, On the enumeration of one-factorizations of complete graphs containing prescribed automorphism groups, *Math. of Comp.*, to appear.

59 D. P. Shaver, "Construction of (v, k, λ) - configuration using a non-enumerative search technique", Ph. D. Thesis, Syracuse University, 1973.

60 D. R. Stinson, The existence of Howell designs of odd side, *Journal of Comb. Theory* A 32 (1982), 53-65.

61 D. R. Stinson, Hill-climbing algorithms for the construction of combinatorial designs, *Annals of Discrete Math.* 26 (1985), 321-334.

62 D. R. Stinson, Room squares with maximum empty subarrays, *Ars Combinatoria* 20 (1985), 159-166.

63 D. R. Stinson and H. Ferch, 2000000 Steiner triple systems of order 19, *Math. of Comp.* 44 (1985) 533-535.

64 M. Tompa, "Hill-climbing: a feasible search technique for the construction of combinatorial configurations", M. Sc. Thesis, University of Toronto, 1975.

65 W. D. Wallis, On one-factorizations of complete graphs, *Journal Austral. Math. Soc.* 16 (1973), 167-171.

66 W. D. Wallis, A. P. Street and J. S. Wallis, "Combinatorics: Room squares, sum-free sets, Hadamard matrices," Lecture Notes in Math., Vol. 292, Springer, New York, 1972.