ON 1-FACTORIZATIONS OF THE COMPLETE GRAPH AND THE RELATIONSHIP TO ROUND ROBIN SCHEDULES

ERIC NEIL GELLING

ON 1-FACTORIZATIONS OF THE COMPLETE GRAPH AND THE RELATIONSHIP TO ROUND ROBIN SCHEDULES

by

ERIC NEIL GELLING

B.A., University of Victoria, 1968

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ARTS
in the Department
of
Mathematics

ACCEPTED
FACULTY OF GRADUATE STUDIES
DEAN
DAT
/73

We accept this thesis as conforming
to the required standard

UNIVERSITY OF VICTORIA

JULY 1973

Supervisor: Dr. R. E. Odeh

ABSTRACT

The following new results concerning 1-factorizations of the complete graph are proved: (1) There are exactly 6 equivalence classes of 1-factorizations of the complete graph with 8 vertices. (2) There are exactly 396 equivalence classes of 1-factorizations of the complete graph with 10 vertices. Representatives of each of the equivalence classes are presented. The size of the automorphism group of each equivalence class of 1-factorizations of the complete graph with $2n$ vertices for $n \leq 5$ is also found.

Several theorems and results related to 1-factorizations of the complete graph are presented, and the relationship to round robin schedules is shown. An application problem demonstrates the importance of the choice of the equivalence class of round robin schedules in the solvability of the problem.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. R. E. Odeh, and to Dr. D. J. Miller for their guidance, encouragement and endless patience throughout the preparation of this thesis. I would also like to thank the faculty, staff and fellow graduate students of the Department of Mathematics and the staff of the University of Victoria Computing Centre for their time and assistance. Finally, I would like to convey a special thank you to my mother, Mrs. M. P. Gelling, for her careful typing of the manuscript.

CHAPTER 1

INTRODUCTION

In recent years the study of graph theory has been playing an increasingly important role in solving problems both mathematical and non-mathematical in nature. Up to about ten years ago, however, very little graph theory had been studied. Ore [17], in 1963, states in his bibliography that "the number of books on graph theory is very small" and then lists a mere nine of them, five of which are foreign works. Since he wrote those words a vast amount of material has been written on the subject. The computer has provided a powerful tool to graph theorists and much of the recent literature has been a result of the ability to use the computer to exhaustively generate and study characteristics of graphs. The potential for future study seems without limit.

It is the purpose of this thesis to study a particular area of graph theory: the 1-factorization of the complete graph with $2n$ vertices. The initial motivation for the study came from a specific problem of scheduling for a sporting event. It was desired to create a schedule with $2n$ teams such that every team plays every other team exactly once (known as a round robin). Furthermore, the games were to be played on n fields such that no team played more than twice on the same field nor twice in a row on the same field. The problem is described in more detail in Chapter 4, along with a solution for 8 teams and 4 fields. In trying to solve the particular problem, the question arose whether all round robin schedules for $2n$ teams are the same except for re-labelling of teams and rounds. It was discovered that the answer is "yes" for $n \leq 3$ and "no" for $n \geq 4$.

In this study we have identified 6 equivalence classes for round robin schedules with 8 teams and 396 equivalence classes for round robin schedules with 10 teams. The problem is identical to the finding of the equivalence classes of 1-factorizations of the complete graph, and the relationship of the two problems is brought forth in Chapters 2 and 3.

In Chapter 2 we present a summary of graph theory beginning with the basic definitions and proceeding through some theorems which we use in the next chapter. Chapter 3 presents the main results of our study. Here we prove that there exists exactly 6 equivalence classes of 1-factorizations of the complete graph with 8 vertices; and exactly 396 equivalence classes of 1-factorizations of the complete graph with 10 vertices. In addition, we establish the size of the automorphism group for every equivalence class. To conclude the chapter we briefly present a method to produce a representative from each equivalence class of 1-factorizations of the complete graph with 12 vertices. As noted earlier, in Chapter 4 we solve the problem of scheduling for a sporting event. In particular, we demonstrate that, when given certain constraints, the choice of the equivalence class of the round robin schedule can be vital as to whether the problem is solvable or not solvable.

In Appendix A we list representatives and characteristics of each of the equivalence classes of 1-factorizations of the complete graph with 10 vertices. Finally, in Appendix B we give a listing of the relevant computer programs.

CHAPTER 2

GRAPH THEORY

2-1. Introduction

It is the purpose of this section to introduce the terminology of graph theory that will be used later in the study. Most of the definitions are from Harary [8], but a few are from other works such as Wilson [24] and Behzad and Chartrand [2].

By a graph X we will mean a finite non-empty set V with p elements together with a finite set E of q unordered pairs of distinct elements of V. The elements of V are called vertices and the elements of E are called edges. We will frequently write $V(X)$ for the vertex set of the graph X and $E(X)$ for the edge set.

A labelling of the graph X with p vertices is a one-to-one mapping from $V(X)$ onto the set $\{0,1,2, \ldots, p-1\}$. We will usually denote the vertex set $V(X)$ as $\{0,1,2, \ldots, p-1\}$ and the edge set $E(X)$ as $\{[x,y]: x, y \in V(X), x \neq y\}$. The number of vertices in $V(X)$ (or the cardinality of $V(X)$) will be represented by $|V(X)|$. Similarly the cardinality of $E(X)$ will be $|E(X)|$.

If $e = [x,y] \in E(X)$ then e is incident with both x and y, and x and y are both incident with e. Two vertices (edges) joined by an edge (vertex) are said to be adjacent.

If X and Y are graphs such that $V(Y) \subseteq V(X)$ and $E(Y) \subseteq E(X)$ then $Y \subseteq X$ and Y is called a <u>subgraph</u> of X. If $V \subseteq V(X)$ then the subgraph $\langle V \rangle$ <u>induced by V</u> is the maximal subgraph of X whose vertex set is V; i.e., it is the graph having vertex set V and whose edge set consists of those edges in X incident with two vertices of V. Similarly, if $E \subseteq E(X)$ then the subgraph $\langle E \rangle$ <u>induced by E</u> is the graph whose edge set is E and whose vertex set consists of those vertices of $V(X)$ incident with at least one edge of E. A subgraph of X is called a <u>spanning</u> subgraph if it contains all the vertices of X.

A <u>walk</u> of the graph X is an alternating sequence of vertices and edges $\{x_o,[x_o,x_1], x_1, [x_1,x_2], x_2, \dots x_{m-1}, [x_{m-1},x_m], x_m\}$ beginning and ending with vertices in which each edge is incident with the vertices immediately preceding and following it. In order to shorten the notation we will represent the walk by the sequence of vertices $\{x_o,x_1,\dots,x_m\}$ with the understanding that $[x_i,x_{i+1}]$ for $0 \leq i \leq m-1$ are the edges of the walk. A <u>trail</u> of the graph X is a walk such that the edges are distinct. A <u>path</u> of the graph X is a trail $\{x_o,x_1, \dots, x_m\}$ such that all vertices are distinct with the possible exception of x_o and x_m. If $x_o = x_m$ the path is closed and is called a <u>circuit</u>.

The <u>order</u>, k, of a circuit C is the number of edges (or vertices) in C. The circuit C, then, will be called a k-circuit. It is clear that if the graph X has p vertices and $C \subseteq X$ then $3 \leq k \leq p$. If $C \subseteq X$ and the order of C is p then C spans the vertices of X and is called a Hamiltonian circuit. A graph with a Hamiltonian circuit is known as a Hamiltonian graph.

A graph is connected if every two vertices are joined by a path. A component of X is a maximal connected subgraph of X. Thus, a connected graph has only one component; a disconnected graph has more than one component.

Let X be a graph with p vertices with the condition that every pair of its p vertices is adjacent. Then X is called a complete p-graph and is denoted by K_p (after Kuratowski). We observe that the number of edges in K_p is given by:

$$|E(K_p)| = \frac{p(p-1)}{2} .$$

The complement $\overline{X}$ of X is a graph which has $V(X)$ as the vertex set and in which two vertices are adjacent if and only if they are not adjacent in X, i.e.

$$V(\overline{X}) = V(X)$$
$$[x,y] \in E(\overline{X}) \text{ iff } x \neq y \text{ and } [x,y] \notin E(X) .$$

The degree of the vertex x is the number of edges incident with it. From this definition comes the first theorem of graph theory, known by Euler over two hundred years ago.

THEOREM 2.1. The sum of the degrees of the vertices of a graph X is twice the number of edges, i.e.,

$$\sum_{x \in V(X)} \deg(x) = 2|E(X)| \; .$$

This theorem is commonly called the "handshaking lemma" -- if several people shake hands, the total number of hands shaken must be even, precisely because there are always two hands to a shake. A natural corollary follows from this theorem.

COROLLARY 2.2. In any graph X, the number of vertices of odd degree is even.

If all vertices of X have the same degree, r, then X is called regular of degree r. A regular graph of degree 0 contains no edges; a regular graph of degree 1 has components consisting of exactly one edge. It is clear that the complete graph K_p is regular of degree $p-1$. It is also clear that the complement of a regular graph is regular.

2-2. 1-Factors, 1-Factorizations and Round Robin Schedules

In this section we present some definitions and theorems which will form a basis for the study of round robin schedules. A round robin schedule involving $2n$ teams has the property that every team plays every other team exactly once. In a round robin schedule there are $2n-1$ <u>rounds</u> of n <u>games</u> each. We will define 1-factors and a 1-factorization of the complete graph K_{2n} and show the relationship to round robin schedules. In addition we shall present some theorems characteristic of 1-factors of a graph.

A <u>factor</u> X_i of a graph X is a spanning subgraph of X which is not totally disconnected. The set of graphs $\{X_1, X_2, \ldots, X_q\}$ is a <u>decomposition</u> of the graph X into the factors $X_1, X_2, \ldots, X_q$ if and only if the following are true:

i. $V(X_i) = V(X)$ for $1 \leq i \leq q$

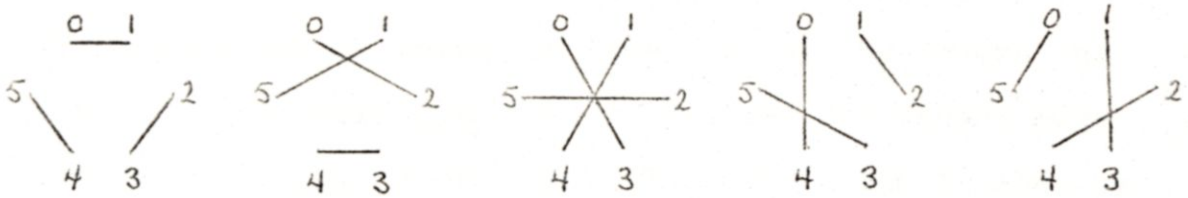
ii. $E(X_i) \cap E(X_j) = \phi$ for all $i \neq j$.

iii. $E(X_1) \cup E(X_2) \cup \ldots \cup E(X_q) = E(X)$.

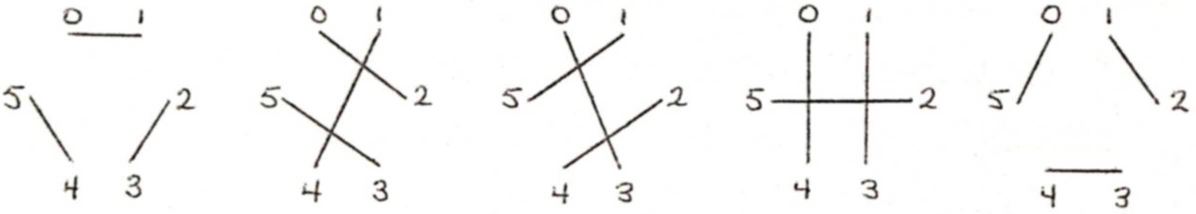
An <u>n-factor</u> is a factor which is regular of degree n. If every X_i of the decomposition $\{X_1, X_2, \ldots, X_q\}$ of the graph X is an n-factor then the decomposition is called an <u>n-factorization</u>, and X itself is <u>n-factorable</u>. Two examples of 1-factorizations of K_6 are given in Figure 2.1.

A famous theorem of Tutte [21] characterizes graphs possessing a 1-factor. Here, an <u>odd component</u> is a component of the graph X with an odd number of vertices.

FIGURE 2.1 TWO 1-FACTORIZATIONS OF K_6



Example 1



Example 2

THEOREM 2.3. *A graph* X *with* p *vertices has a 1-factor if and only if* p *is even and there is no subset* $V \subseteq V(X)$ *such that the number of odd components of* $< V(X) - V >$ *exceeds* $|V|$.

In this study we are interested in the 1-factorizations of the complete graph K_{2n}. Clearly (by observation or by Tutte's Theorem) K_{2n} contains a 1-factor with n edges.

THEOREM 2.4. *The complete graph* K_{2n} *has* $t = (2n-1)(2n-3) \ldots (1)$ *different 1-factors*.

PROOF. Every 1-factor of K_{2n} covers the vertices and has n edges. By fixing one vertex of the 2n vertices, the first edge can be chosen in (2n-1) ways. By fixing one vertex of the remaining (2n-2) vertices the second edge can be chosen in (2n-3) ways. The conclusion follows by induction on the n edges. ∎

COROLLARY 2.5. Let the vertices of K_{2n} be $\{0,1, \ldots, 2n-1\}$ and let W be the set of t 1-factors of the graph K_{2n}. Let $W_i \subseteq W$ contain all the 1-factors in W with edge $[0,i]$ for $i = 1, \ldots, 2n-1$. Then $W_1, \ldots, W_{2n-1}$ represents a partition of W into $2n-1$ subsets with $s = (2n-3)(2n-5) \ldots (1)$ members each.

PROOF. Follows from the proof of Theorem 2.4. ∎

The fact that K_{2n} is 1-factorable is well known, and schemes for generating a 1-factorization (which we will show is equivalent to forming a round robin schedule) have been known since at least 1859 (Reisz [19]). Schemes are given in Ore [17], Moon [16], König [10], Freund [6], Lockwood [12] and [13], and Kraitchik [11]. The particular proof below is based on Harary [8].

THEOREM 2.6. The complete graph K_{2n} is 1-factorable.

PROOF. By definition, K_{2n} has $2n$ vertices and any 1-factor in K_{2n} has n edges. To be 1-factorable, the graph K_{2n}, which has $n(2n-1)$ edges, must be partitionable into $(2n-1)$ 1-factors. Let the vertices of K_{2n} be $\{0,1,2, \ldots, 2n-1\}$. Let $B = \{1,2, \ldots, 2n-1\}$. Then for $\alpha \varepsilon B$ consider the edge-induced subgraph X_α of K_{2n} defined as follows:

$$E(X_\alpha) = \{[0,\alpha] \cup ([\alpha+\mu, \alpha-\mu]: \quad \mu = 1,2, \ldots, n-1)\}$$

where each of the $\alpha+\mu$, $\alpha-\mu$ is an element of B modulo $(2n-1)$. A diagram of the subgraph is shown in Figure 2.2, where the vertices in B are on the circle, while the vertex 0 is outside the circle.

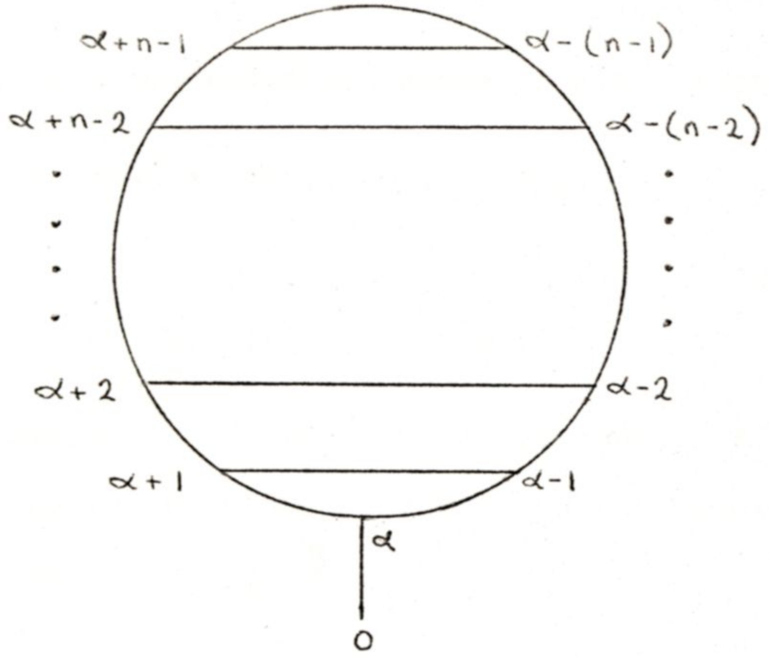
We claim that X_α is a 1-factor of K_{2n}. Clearly, since $0 \notin B$, the vertex 0 appears only once in $E(X_\alpha)$. Also, since $B - \{\alpha\}$ contains exactly $2(n-1)$ elements it is clear that the vertex α appears only once (i.e., in the edge $[0,\alpha]$) and that in fact every vertex appears exactly once. Therefore, X_α is a 1-factor for each $\alpha \varepsilon B$.

We claim that for $\beta \neq \alpha$ and $\beta, \alpha \varepsilon B$, X_β and X_α are edge disjoint. Clearly, $[0,\beta] \neq [0,\alpha]$. Assume

$$[\beta+\omega, \beta-\omega] = [\alpha+\mu, \alpha-\mu], \; 1 \leq \mu,\omega \leq n-1$$

where each vertex is contained in B modulo $2n-1$.

FIGURE 2.2 A 1-FACTORIZATION OF K_{2n} BY THEOREM 2.6.



Without loss of generality, assume

$$\beta+\omega \equiv \alpha+\mu \pmod{2n-1}.$$

Then

$$\beta-\omega \equiv \alpha-\mu \pmod{2n-1}.$$

By adding we get

$$2\beta \equiv 2\alpha \pmod{2n-1}$$

and since $2n-1$ is odd we get

$$\beta \equiv \alpha \pmod{2n-1}.$$

But this is a contradiction since $\beta \neq \alpha$. Therefore

$$[\beta+\omega, \beta-\omega] \neq [\alpha+\mu, \alpha-\mu], \quad 1 \leq \mu,\omega \leq n-1$$

and

$$E(X_\alpha) \cap E(X_\beta) = \phi .$$

Since B contains $2n-1$ elements, $X_1 \cup X_2 \cup \ldots \cup X_{2n-1}$ contains $n(2n-1)$ edges and represents a complete 1-factorization of K_{2n}. ∎

A round robin schedule on $2n$ vertices (hereafter called simply a schedule) is a 1-factorization of the complete graph K_{2n}. The vertices of K_{2n} correspond to the teams, the edges correspond to the games, and the 1-factors correspond to the rounds. We shall use the terms "1-factorization" and "schedule" interchangeably in our study.

The following set of theorems and definitions characterize the nature of 1-factors in a graph.

THEOREM 2.7. *Let* X_i *and* X_j *be two disjoint 1-factors of any graph* X. *Then every vertex of* X *has exactly one edge of* X_i *and exactly one edge of* X_j *incident with it*.

PROOF. Since each 1-factor of X covers the vertices and since $E(X_i) \cap E(X_j) = \phi$, the theorem follows immediately. ■

In any graph X, an *alternating trail* is a trail such that the edges alternate between two disjoint 1-factors of X.

COROLLARY 2.8. *Let* X_i *and* X_j *be two disjoint 1-factors of any graph* X. *Then, starting at any vertex of* X, *say* x_o, *there exists an alternating trail which eventually returns to* x_o. *Furthermore, the first vertex the alternating trail passes through twice is* x_o, *and thus* x_o *lies on a circuit*.

PROOF. Follows immediately from Theorem 2.7. ■

In any graph X, an *alternating circuit* is a circuit such that the edges alternate between two disjoint 1-factors of X. If the alternating circuit covers the vertices of X then it is an *alternating Hamiltonian circuit*.

COROLLARY 2.9. *Let* X_i *and* X_j *be two disjoint 1-factors of any graph* X. *Then every vertex of* X *is a member of an alternating circuit between* X_i *and* X_j.

PROOF. It follows from Corollary 2.8 that every vertex lies on a circuit. But by Theorem 2.7, this circuit must be an alternating circuit. ■

It will be observed that, by the definition of an alternating circuit, the length k of an alternating circuit is even and furthermore $k \geq 4$.

COROLLARY 2.10. *Let* X_i *and* X_j *be two disjoint 1-factors of any graph* X. *Then* $X_i \cup X_j$ *is the disjoint union of alternating circuits*.

PROOF. Follows immediately from Corollary 2.9. ■

The next theorem and corollary are interesting results of Theorem 2.6. Let

$$F = \{Y_i \; ; \; 1 \leq i \leq 2n-1\}$$

be a 1-factorization of the complete graph K_{2n}. Then F is a *completely Hamiltonian schedule* (or an *H-schedule*) if, for every $Y_i, Y_j \in F$ and $i \neq j$, $Y_i \cup Y_j$ is an alternating Hamiltonian circuit. A completely non-Hamiltonian schedule (or an NH-schedule) is a schedule which has no Hamiltonian circuits between disjoint 1-factors.

THEOREM 2.11. Let K_{2n} be partitioned as in Theorem 2.6 and let X_α and $X_{\alpha+\beta}$ be two disjoint 1-factors defined by this partitioning. Then $X_\alpha \cup X_{\alpha+\beta}$ is an alternating Hamiltonian circuit if and only if there exists no integer c, $0 < c < 2n-1$, such that $c\beta \equiv 0 \pmod{2n-1}$.

PROOF. X_α and $X_{\alpha+\beta}$ are defined in Theorem 2.6 as follows:

$$E(X_\alpha) = \{[0,\alpha] \cup ([\alpha+\mu, \alpha-\mu]: \quad \mu = 1,2, \ldots, n-1)\}$$

$$E(X_{\alpha+\beta}) = \{[0,\alpha+\beta] \cup ([\alpha+\beta+\mu, \alpha+\beta-\mu]: \quad \mu = 1,2, \ldots, n-1)\}$$

where each of the α, $\alpha+\beta$, $\alpha+\mu$, $\alpha-\mu$, $\alpha+\beta+\mu$, $\alpha+\beta-\mu$ is an element of $\{1,2, \ldots, 2n-1\}$ modulo $(2n-1)$. See Figure 2.3 for examples where $n = 5$ and $\beta = 3$ and 2.

By the definition of X_α and $X_{\alpha+\beta}$ and by Corollary 2.9 there exists an alternating circuit on $X_\alpha \cup X_{\alpha+\beta}$:

$$\{0, \alpha, \alpha+2\beta, \alpha-2\beta, \alpha+4\beta, \alpha-4\beta, \ldots, \alpha+x\beta, \alpha-x\beta, 0\}.$$

Therefore for some even x, there exists an alternating circuit which includes the vertex o and which must contain $x+2$ vertices. However, since the edge $[0,\alpha+\beta] \in E(X_{\alpha+\beta})$

must be in the alternating circuit, we have

$$\alpha - x\beta \equiv \alpha + \beta \pmod{2n-1}.$$

Let $c = x+1$. Then

$$\alpha - x\beta = \alpha - (c-1)\beta \equiv \alpha + \beta \pmod{2n-1}$$

and hence

$$c\beta \equiv 0 \pmod{2n-1} .$$

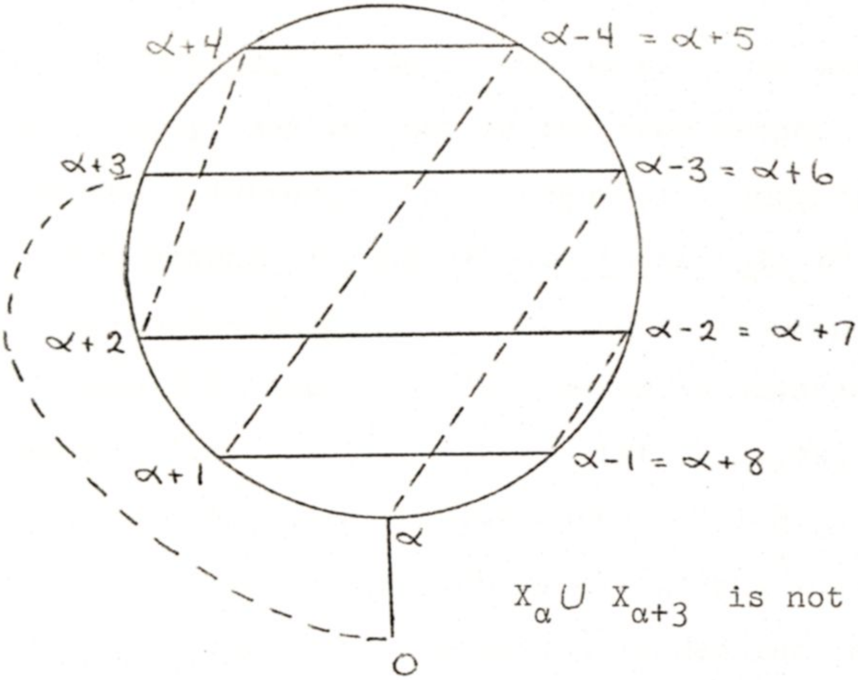
But since $X_\alpha \cup X_{\alpha+\beta}$ is an alternating Hamiltonian circuit if and only if $x + 2 = 2n$, the conclusion follows. ∎

COROLLARY 2.12. <u>The schedule which is defined by the partitioning of</u> K_{2n} <u>in Theorem 2.6 is an H-schedule if and only if</u> $2n-1$ <u>is a prime number</u>.

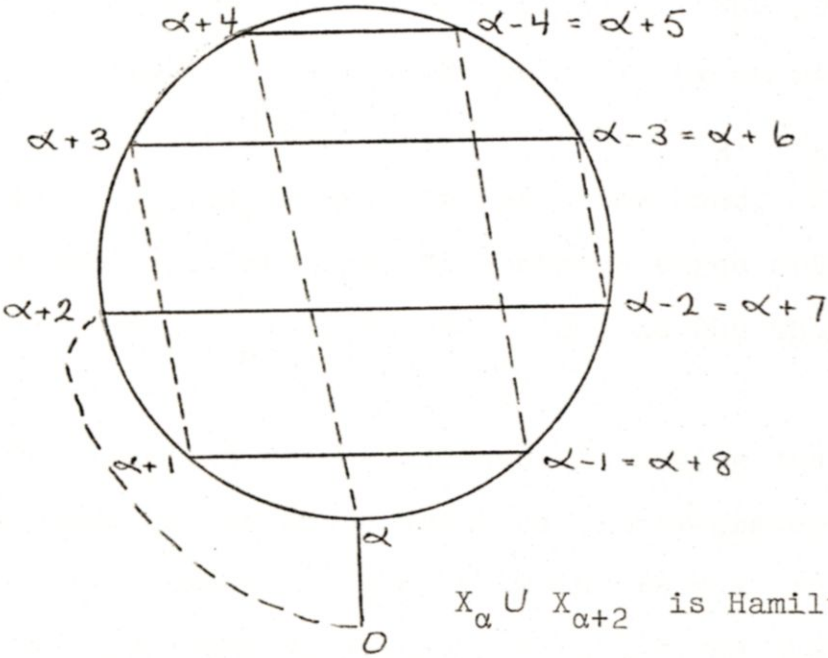
By Corollary 2.12 we observe that the schedules defined by Theorem 2.6 for K_4, K_6, K_8, K_{12}, K_{14}, K_{18}, K_{20}, K_{24} must be H-schedules. Similarly, the schedules for K_{10}, K_{16}, K_{22}, K_{26} must not be H-schedules.

An interesting unsolved question is whether or not for every $n \geq 2$ there exists an H-schedule and for every $n \geq 4$ whether or not there exists an NH-schedule.

FIGURE 2.3 ILLUSTRATION OF THEOREM 2.11 FOR 1-FACTORS OF K_{10}.



$X_\alpha \cup X_{\alpha+3}$ is not Hamiltonian



$X_\alpha \cup X_{\alpha+2}$ is Hamiltonian

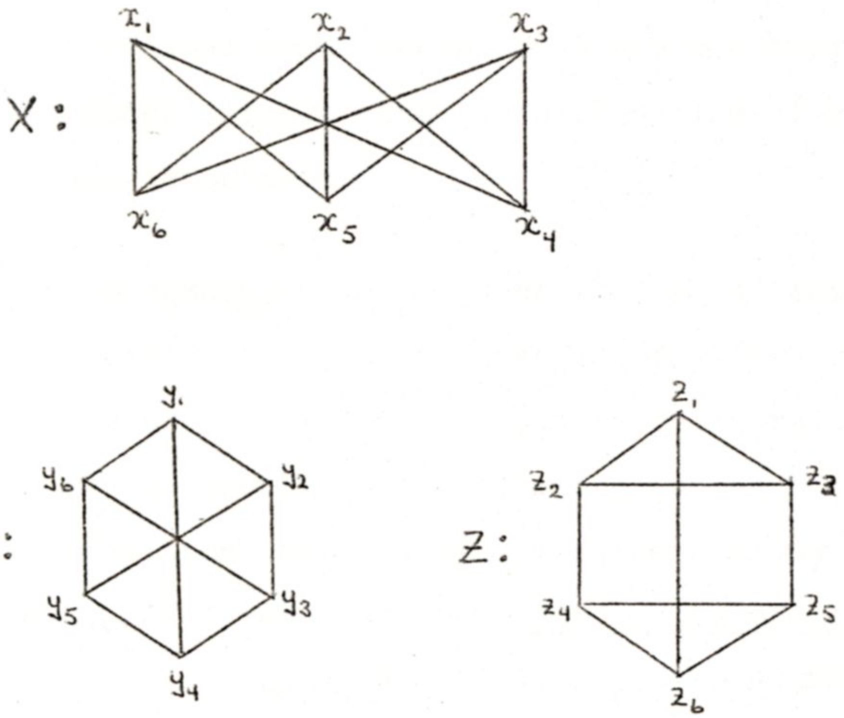
2-3. Isomorphisms, Automorphism Groups, and Equivalence Classes

Two graphs, X and Y, which may look quite different geometrically, may in fact be the same graph. For this purpose we shall introduce the concept of isomorphism.

Two graphs X and Y are isomorphic if there exists a one-to-one mapping ψ, called an isomorphism, from V(X) onto V(Y) such that ψ preserves adjacency and nonadjacency; i.e. $[x_1,x_2] \varepsilon E(X)$ iff $[\psi x_1,\psi x_2] \varepsilon E(Y)$. Clearly, isomorphism is an equivalence relation on graphs, and hence divides the collection of all graphs into equivalence classes. If there exists no mapping on V(X) which will map X onto Y then X and Y are non-isomorphic and belong to different equivalence classes. The graphs X and Y of Figure 2.4 are isomorphic by the mapping $\psi : V(X) \rightarrow V(Y)$ defined by: $\psi x_1 = y_1$, $\psi x_2 = y_3$, $\psi x_3 = y_5$, $\psi x_4 = y_2$, $\psi x_5 = y_4$, $\psi x_6 = y_6$. On the other hand, X and Z are non-isomorphic because Z contains three mutually adjacent vertices but X does not. (Behzad and Chartrand [2], p. 3).

Two labelled graphs X and Y having the same labels are identical if there exists a (label-preserving) isomorphism ψ between X and Y with $\psi x = x$ for all $x \varepsilon V(X)$. By referring to Figure 2.4 it will be observed that X and Y are isomorphic but not identical.

FIGURE 2.4 AN ILLUSTRATION OF ISOMORPHISM AND NON-ISOMORPHISM.



The following two theorems are obvious results of the definitions.

THEOREM 2.13. *If* X_i *and* X_j *are 1-factors of the graph* X *then* X_i *and* X_j *are isomorphic.*

THEOREM 2.14. *If* C_i *and* C_j *are two circuits of length* k *then* C_i *and* C_j *are isomorphic*.

It follows from Theorem 2.14 and the definition of an alternating circuit that two alternating circuits of length k are isomorphic.

An *automorphism* of a graph X is an isomorphism of X with itself, i.e., a permutation of $V(X)$ which preserves adjacency. Clearly, an automorphism followed by another is an automorphism and therefore (under the operation of composition) the set of all automorphisms on X form a group, $G(X)$, known as the *automorphism group of X*. The *order* of $G(X)$, denoted $|G(X)|$, is the number of members in the group. Since the identity mapping on X is always in $G(X)$, $|G(X)| \geq 1$. The identity mapping is the trivial automorphism of X; all other automorphisms, if any exist, are non-trivial. Using Tutte's definition [22, p. 52] a graph is *symmetrical* if it has a non-trivial automorphism.

THEOREM 2.15. *For any graph* X *and its complement* $\overline{X}$, $G(X)$ *is isomorphic to* $G(\overline{X})$.

PROOF. Every element ψ of $G(X)$ is a permutation on $V(X)$ which preserves adjacency in X. However, ψ preserves adjacency if and only if ψ preserves non-adjacency. Thus a permutation on $V(X)$ is an automorphism of X if and only if it is an automorphism on $\overline{X}$, implying that $G(X)$ is isomorphic to $G(\overline{X})$. ∎

Several results follow from the above definitions.

Result 1: The order of K_p is $p!$ because a mapping from any vertex to any other vertex will preserve the structure of K_p.

Result 2: The order of a circuit of length k is $2k$. To show this let a circuit be represented by $C = \{x_o, x_1, \ldots, x_k\}$ where $x_k = x_o$ and the subscripts are modulus k. Each subscript of the vertices can be increased by 1 without changing the circuit, thus transforming C into one of its rotations. Each subscript i can also be replaced by $k-i$, thus transforming C into its mirror image. Since there are k rotations and a mirror image of each, the order of C is $2k$.

Result 3: The order of a 1-factor with $2n$ vertices is $n!2^n$. To show this we observe that since there are n disjoint edges they can be permuted $n!$ ways. But since there are two vertices incident with each edge, there are 2^n ways of permuting the vertices on the edges.

Result 4: The order of a 1-factor in a 1-factorization of K_{2n} is $(2n-1)n!2^n$. By result 3 above, the order of a 1-factor is $n!2^n$. But since a 1-factorization of K_{2n} has (2n-1) 1-factors and (by Theorem 2.13) they are isomorphic, every 1-factor can be mapped onto every other 1-factor. Therefore, if X_i is a 1-factor in a 1-factorization of K_{2n}, there are $(2n-1)n!2^n$ mappings on the vertices of K_{2n} which will preserve X_i.

THEOREM 2.16. The number of ways in which a graph X with p vertices can be labelled is $\frac{p!}{|G(X)|}$.

PROOF. (Behzad and Chartrand [2], p. 167). Let $\{x_0, \ldots, x_{p-1}\}$ be a set of p labels. Clearly there are p! ways of labelling X without regard to the number of resulting labelled graphs which may be identical. Let X_1 and X_2 be two labelled graphs obtained from X. Then the relation "X_1 is identical to X_2" is an equivalence relation on the set of labelled graphs obtained from X. For the labelled graph X_1, an automorphism of X gives rise to a labelled graph identical to X_1, and conversely. Hence, each equivalence class of a labelling X_1 of X contains $|G(X)|$ members. This implies that there are $p!/|G(X)|$ labellings in all. ∎

Throughout this study we will let $L(X)$ represent the set of all labellings of X and $|L(X)|$ represent the number of elements in $L(X)$.

One of the more difficult problems of graph theory has been to find an easy test for isomorphism between graphs. Much research has been devoted by chemists, network analysts, linguists, logicians, etc., to solve this problem. Probably the most efficient procedure to date was developed by Corneil and Gotlieb in 1970 [5], but unfortunately it is based on a conjecture. It remains still unsolved to find an efficient deterministic algorithm (i.e. one which can be solved in time T where T is proportional to a constant power of the number of vertices of the graph) to determine whether two graphs are isomorphic.

CHAPTER 3

EQUIVALENCE CLASSES OF 1-FACTORIZATIONS OF THE COMPLETE GRAPH K_{2n}

3-1. Introduction

In this chapter, vertices will be labelled from the set $\{1,2, \ldots, 2n\}$ in order to correspond to the usual labellings of teams in round robin schedules.

Throughout this chapter let the set W be the set of 1-factors of K_{2n}. For $i = 1, \ldots, 2n-1$ we will let W_i represent the set of all 1-factors in W with the edge $[1, i+1]$. By Corollary 2.5, W will thus be partitioned into $2n-1$ sets, W_i for $i = 1, \ldots, 2n-1$, each containing $s = (2n-3)(2n-5)\ldots(1)$ 1-factors.

We observe that any 1-factorization of K_{2n} will have exactly one member from each W_i.

From Theorem 2.13 we know that all 1-factors of K_{2n} are isomorphic. Let $X_1, X_2 \varepsilon W_1$ such that $X_1 \neq X_2$. Let T_1 and T_2 be the complete sets of 1-factorizations of K_{2n} which contain X_1 and X_2 respectively. Then a mapping ψ which maps X_1 onto X_2 must also be a one-to-one mapping of T_1 onto T_2. Therefore, in searching for equivalence classes of 1-factorizations of K_{2n} and in enumerating the members in each class, we can reduce our search by fixing a particular 1-factor $X_1 \varepsilon W_1$ to be a member of every 1-factorization studied.

Without loss of generality, we will let X_1 be the 1-factor whose edge set is $\{[1,2], [3,4], \ldots, [2n-1, 2n]\}$.

In this chapter we will study the 1-factorizations of K_{2n} for $n \leq 5$. We observe immediately that the 1-factorizations of K_2 and K_4 are trivial. The only 1-factors of K_4 are:

$$X_1: \{[1,2], [3,4]\}$$

$$X_2: \{[1,3], [2,4]\}$$

$$X_3: \{[1,4], [2,3]\}$$

Clearly, each W_i for $i = 1,2,3$ contains only one member and therefore $F = \{X_1, X_2, X_3\}$ must be the only 1-factorization of K_4. By Theorem 2.16, the order $|G(F)|$ of the automorphism group of F is $4! = 24$.

By Corollary 2.5 the set of 1-factors, W, for K_6 contains 15 members and each W_i for $i = 1, \ldots, 5$ contains 3 1-factors. They are as follows:

$$W_1\begin{cases} X_1: \{[1,2], [3,4], [5,6]\} \\ X_2: \{[1,2], [3,5], [4,6]\} \\ X_3: \{[1,2], [3,6], [4,5]\} \end{cases} \qquad W_2\begin{cases} X_4: \{[1,3], [2,4], [5,6]\} \\ X_5: \{[1,3], [2,5], [4,6]\} \\ X_6: \{[1,3], [2,6], [4,5]\} \end{cases}$$

$$W_3 \begin{cases} X_7\colon \{[1,4],\ [2,3],\ [5,6]\} \\ X_8\colon \{[1,4],\ [2,5],\ [3,6]\} \\ X_9\colon \{[1,4],\ [2,6],\ [3,5]\} \end{cases} \qquad W_4 \begin{cases} X_{10}\colon \{[1,5],\ [2,3],\ [4,6]\} \\ X_{11}\colon \{[1,5],\ [2,4],\ [3,6]\} \\ X_{12}\colon \{[1,5],\ [2,6],\ [3,4]\} \end{cases}$$

$$W_5 \begin{cases} X_{13}\colon \{[1,6],\ [2,3],\ [4,5]\} \\ X_{14}\colon \{[1,6],\ [2,4],\ [3,5]\} \\ X_{15}\colon \{[1,6],\ [2,5],\ [3,4]\} \end{cases}$$

It can easily be checked that there are only two 1-factorizations of K_6 which include the fixed 1-factor X_1. They are:

$$F_1\colon \{X_1,\ X_5,\ X_9,\ X_{11},\ X_{13}\}$$

$$F_2\colon \{X_1,\ X_6,\ X_8,\ X_{10},\ X_{14}\}$$

The mapping $\psi\colon F_2 \to F_1$ defined by:

$$1 \to 1, \quad 2 \to 2, \quad 3 \to 3, \quad 4 \to 4, \quad 5 \to 6, \quad 6 \to 5$$

shows that there is only one equivalence class, which we shall label $L(F_1)$. Clearly, since W_1 contains 3 members, $L(F_1)$ contains 6 members. By Theorem 2.16, the order of the automorphism group for $L(F_1)$ is $6!/6 = 120$.

3-2. Results for K_8.

By Corollary 2.5 the set of 1-factors, W, for K_8 contains 105 members and each W_i for $i = 1, \ldots, 7$ contains 15 1-factors. A computer program (see Computer Program 1 in Appendix B) was written to generate the set W and find all the 1-factorizations of K_8. The program fixed the first 1-factor $X_1 \in W_1$ and chose 1-factors lexicographically out of each W_i for $i = 2, \ldots, 7$, choosing only 1-factors which did not have a common edge with previously chosen 1-factors. The procedure followed is given by the flowchart and supporting explanation in Figure 3.1, and corresponds to the well-known method of efficient search known as backtrack programming. See Golomb and Baumert [7] or Walker [23] for a general description of backtrack programming.

FIGURE 3.1 FLOWCHART FOR GENERATING SCHEDULES FROM THE PARTITIONED SET OF 1-FACTORS OF K_8 .

Variables

n = number of edges in a 1-factor.

nm = 2n-1 = number of 1-factors in a schedule.

s = (2n-3)(2n-5) ... (1) = number of elements in each W_i.

W(i,k) = k^{th} element in the set W_i for $i = 1, \ldots, nm$ and $k = 1, \ldots, s$.

FIGURE 3.1 (cont'd)

P(i) = current pointer to the 1-factor in each W_i for $i = 1, \ldots, nm$.

ncnt = count of schedules.

SCHED(ncnt,i) = output array for schedules.

Explanation of flowchart

Step 1 - Initialize variables and arrays. (Note that i is initialized to 2 instead of to 1 because the program fixes the first factor in W_1).

Step 2 - Are there any 1-factors left in W_i which are candidates? If yes, go to Step 5.

Step 3 - Are we finished, i.e. are there any candidates left in W_2? If yes, go to Step 10.

Step 4 - Initialize pointer (P_i) in W_i to 1 and backtrack to previous W_i. Pick next candidate in new W_i. Go to Step 2.

Step 5 - Does the current candidate from W_i have edges in common with candidates from W_j, $j = 1, \ldots, i-1$? If no, go to Step 7.

Step 6 - Choose next candidate in W_i and go to Step 2.

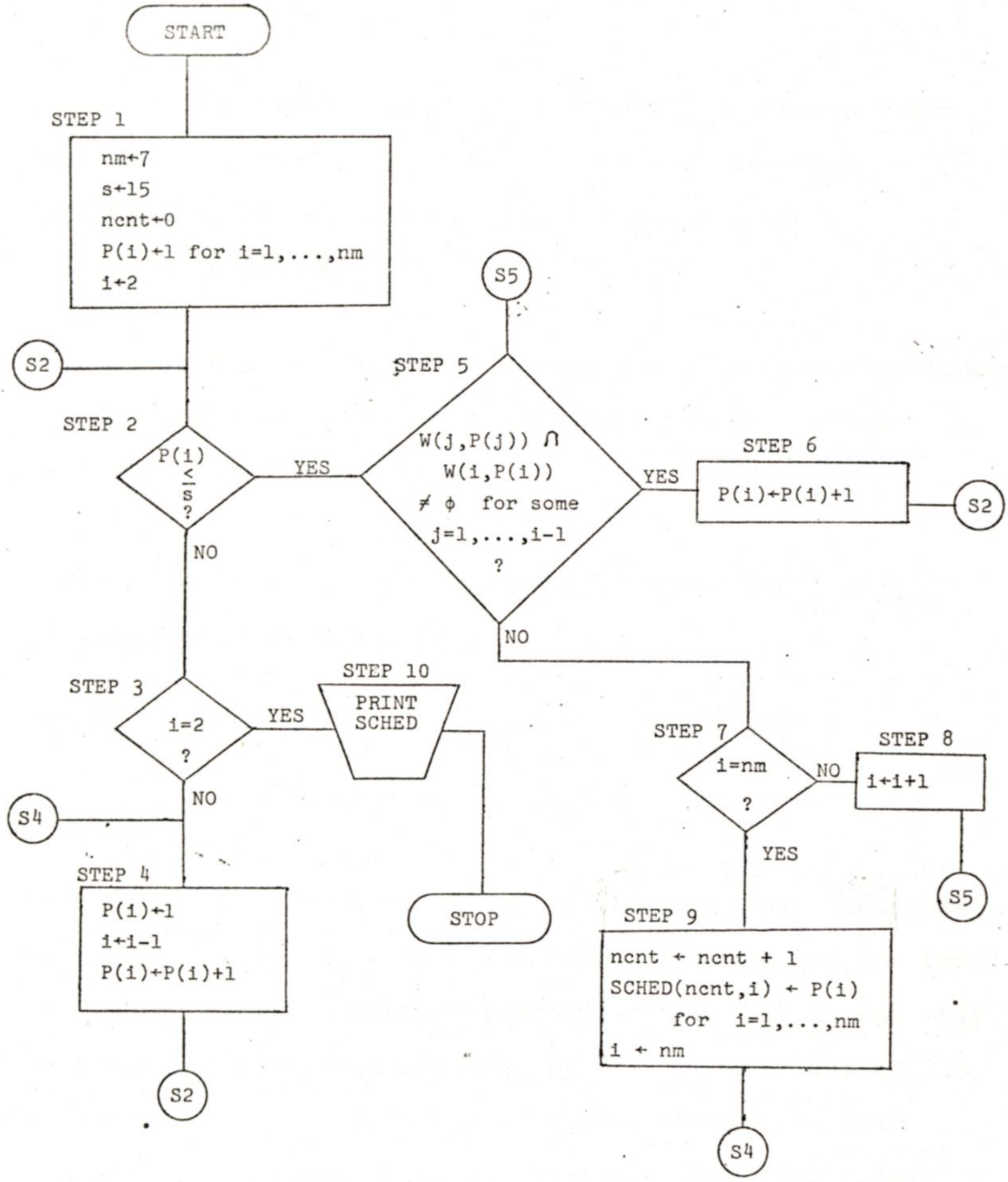
Step 7 - Has a schedule been found? If yes, go to Step 9.

Step 8 - Move forward to next W_i and go to Step 5.

Step 9 - Move schedule to output array. Set i to last W_i in preparation for backtrack and go to Step 4.

Step 10 - Print all schedules and stop.

FIGURE 3.1 (cont'd)
FLOWCHART



THEOREM 3.1. There are 6240 1-factorizations of K_8 labelled on the set $\{1,2, \ldots, 8\}$.

PROOF. Computer Program 1 generated 416 schedules which include the 1-factor $X_1 \varepsilon W_1$. Since W_1 contains 15 1-factors there are 15 sets of 416 schedules or 6240 schedules in all. ∎

To find the equivalence classes of 1-factorizations of K_8 the computer program performed mappings which preserved the fixed 1-factor X_1.

LEMMA 3.2. There are a maximum of 6 equivalence classes of 1-factorizations of K_8 .

PROOF. Let

$$G_1 = \{\psi \varepsilon G(K_8) : \psi X_1 = X_1\} .$$

Then by Result 4 in Section 2-3 of the previous chapter we know that $|G_1| = 2^4 \cdot 4! \cdot 7 = 2688$. However, of the total of 2688 mappings, Computer Program 1 found that it was only necessary to perform 9 mappings on the complete set of 416 schedules with X_1 in order to reduce the set to six equivalence classes. The nine mappings (ignoring trivial mappings of a vertex onto itself) are:

ψ_1: $1 \to 2,\ 2 \to 1$

ψ_2: $3 \to 4,\ 4 \to 3$

ψ_3: $5 \to 6,\ 6 \to 5$

ψ_4: $7 \to 8,\ 8 \to 7$

ψ_5: $5 \to 7,\ 7 \to 5,\ 6 \to 8,\ 8 \to 6$

ψ_6: $3 \to 5,\ 5 \to 3,\ 4 \to 6,\ 6 \to 4$

ψ_7: $1 \to 3,\ 3 \to 1,\ 2 \to 4,\ 4 \to 2$

ψ_8: $2 \to 3,\ 3 \to 2,\ 5 \to 8,\ 8 \to 5$

ψ_9: $2 \to 6,\ 3 \to 2,\ 6 \to 7,\ 7 \to 3$ ∎

Figure 3.2 gives a listing of representatives of the six equivalence classes found by Computer Program 1.

FIGURE 3.2 REPRESENTATIVES FROM EACH OF THE EQUIVALENCE CLASSES OF 1-FACTORIZATIONS OF K_8 .

F_1: $\{X_1, X_2, X_4, X_7, X_{11}, X_{16}, X_{22}\}$

F_2: $\{X_1, X_2, X_4, X_7, X_{11}, X_{17}, X_{21}\}$

F_3: $\{X_1, X_2, X_4, X_{10}, X_{13}, X_{15}, X_{20}\}$

F_4: $\{X_1, X_2, X_4, X_8, X_{12}, X_{17}, X_{19}\}$

F_5: $\{X_1, X_2, X_5, X_9, X_{13}, X_{14}, X_{20}\}$

F_6: $\{X_1, X_3, X_6, X_9, X_{13}, X_{14}, X_{18}\}$

FIGURE 3.2 (cont'd)

The 1-factors $X_1, X_2, \ldots, X_{22}$ are as follows:

X_1: {[1,2], [3,4], [5,6], [7,8]}

X_2: {[1,3], [2,4], [5,7], [6,8]}

X_3: {[1,3], [2,5], [4,7], [6,8]}

X_4: {[1,4], [2,3], [5,8], [6,7]}

X_5: {[1,4], [2,5], [3,8], [6,7]}

X_6: {[1,4], [2,6], [3,8], [5,7]}

X_7: {[1,5], [2,6], [3,7], [4,8]}

X_8: {[1,5], [2,6], [3,8], [4,7]}

X_9: {[1,5], [2,7], [3,6], [4,8]}

X_{10}: {[1,5], [2,7], [3,8], [4,6]}

X_{11}: {[1,6], [2,5], [3,8], [4,7]}

X_{12}: {[1,6], [2,7], [3,5], [4,8]}

X_{13}: {[1,6], [2,8], [3,7], [4,5]}

X_{14}: {[1,7], [2,3], [4,6], [5,8]}

X_{15}: {[1,7], [2,5], [3,6], [4,8]}

X_{16}: {[1,7], [2,8], [3,5], [4,6]}

X_{17}: {[1,7], [2,8], [3,6], [4,5]}

X_{18}: {[1,8], [2,4], [3,5], [6,7]}

FIGURE 3.2 (cont'd)

X_{19}: {[1,8], [2,5], [3,7], [4,6]}

X_{20}: {[1,8], [2,6], [3,5], [4,7]}

X_{21}: {[1,8], [2,7], [3,5], [4,6]}

X_{22}: {[1,8], [2,7], [3,6], [4,5]}

THEOREM 3.3. There are exactly 6 equivalence classes of 1-factorizations of K_8 .

PROOF. Lemma 3.2 proves that there are no more than 6 equivalence classes. However, a count of the number of alternating Hamiltonian circuits in each of the representatives $F_1, \ldots, F_6$ in Figure 3.2 shows that there are 0, 8, 12, 14, 18 and 21 alternating Hamiltonian circuits in the $F_1, \ldots, F_6$ respectively. Clearly, therefore, they have different structure and they are non-equivalent. ▮

A natural corollary follows immediately from this theorem.

COROLLARY 3.4. The number of alternating Hamiltonian circuits in any 1-factorization of K_8 will characterize the 1-factorization with respect to its equivalence class.

THEOREM 3.5. The number of members in each equivalence class of 1-factorizations of K_8 and the order of the automorphism groups are as follows:

Class	$\lvert L(F_i)\rvert$	$\lvert G(F_i)\rvert$
$L(F_1)$	30	1344
$L(F_2)$	630	64
$L(F_3)$	420	96
$L(F_4)$	2520	16
$L(F_5)$	1680	24
$L(F_6)$	960	42

where the F_i, $i = 1, \ldots, 6$, correspond to the 1-factorizations listed in Figure 3.2.

PROOF. The count of the number of members in each $L(F_i)$ with the fixed 1-factor $X_1 \in W_1$ was performed by the computer program. Since there are 15 1-factors in W_1 each of the computer generated counts was multiplied by 15 in order to give the total number, $|L(F_i)|$, of members in each $L(F_i)$. By Theorem 2.16, $|G(F_i)| = 8!/|L(F_i)|$. ∎

3-3. Results for K_{10}

By Corollary 2.5, the set of 1-factors, W, for K_{10} contains 945 members and each W_i for $i = 1, \ldots, 9$ contains 105 1-factors. It was found that Computer Program 1 would use up far too much computer time to find all the 1-factorizations of K_{10}. It became necessary to reduce the search for schedules by reducing both the number of 1-factors to be considered and to put restrictions on the searches in the backtracking technique. The following set of lemmas enabled us to solve the problem.

LEMMA 3.6. For K_{10}, let W_2' be the set of all 1-factors $X_i \in W_2$ such that $E(X_1) \cap E(X_i) = \phi$. Then $X_1 \cup X_i$ is isomorphic to either $X_1 \cup X_2$ or $X_1 \cup X_3$ where X_2 and X_3 are given in Figure 3.3. There are 20 members in W_2' which, in union with X_1, are isomorphic to $X_1 \cup X_2$. There are 48 members in W_2' which, in union with X_1, are isomorphic to $X_1 \cup X_3$.

PROOF. Of the 105 1-factors in W_2, only 68 do not have an edge in common with X_1 and therefore W_2' has 68 members. By Corollary 2.10, $X_1 \cup X_i$ for $X_i \in W_2'$ is the disjoint union of alternating circuits. Clearly $X_1 \cup X_i$ can only be a Hamiltonian circuit or the disjoint union of a

4-circuit and a 6-circuit. But by Theorem 2.14, all k-circuits are isomorphic. Therefore for every $X_i \varepsilon W_2'$, $X_1 \cup X_i$ must be isomorphic to either $X_1 \cup X_2$ (which is a 4-circuit and a 6-circuit) or $X_1 \cup X_3$ (which is a Hamiltonian circuit). By testing all 68 members in W_2', it was found that 48 of them form a Hamiltonian circuit with X_1 and the remaining 20 form a 4-circuit and a 6-circuit. ∎

FIGURE 3.3 REPRESENTATIVE 1-FACTORS OF K_{10} WITH MULTIPLICATION FACTORS FOR FINDING SIZES OF EQUIVALENCE CLASSES OF 1-FACTORIZATIONS.

Key

M1	# 1-factors in W_1 isomorphic to X_1.
M2	# 1-factors in W_2 which, in union with X_1, are isomorphic to $X_1 \cup X_i$ for $i = 2,3$.
M3	# 1-factors in W_3 which, in union with $X_1 \cup X_2$, are isomorphic to $X_1 \cup X_2 \cup X_i$ for $i = 5,7,8, \ldots, 12$ such that X_1 and $X_1 \cup X_2$ are preserved.
M4	# 1-factors in W_3 which, in union with $X_1 \cup X_3$, are isomorphic to $X_1 \cup X_3 \cup X_i$ for $i = 4, \ldots, 7,13,\ldots, 36$ such that X_1 and $X_1 \cup X_3$ are preserved.

FIGURE 3.3 (cont'd)

1-Factor	Edges [x,y]	M1	M2	M3	M4
X_1	{[1,2], [3,4], [5,6], [7,8], [9,10]}	105			
X_2	{[1,3], [2,4], [5,7], [6,9], [8,10]}		20		
X_3	{[1,3], [2,5], [4,7], [6,9], [8,10]}		48		
X_4	{[1,4], [2,3], [5,7], [6,10], [8,9]}				2
X_5	{[1,4], [2,3], [5,8], [6,10], [7,9]}			3	3
X_6	{[1,4], [2,3], [5,9], [6,8], [7,10]}				2
X_7	{[1,4], [2,3], [5,10], [6,8], [7,9]}			1	1
X_8	{[1,4], [2,5], [3,6], [7,10], [8,9]}			6	
X_9	{[1,4], [2,5], [3,7], [6,10], [8,9]}			6	
X_{10}	{[1,4], [2,5], [3,8], [6,10], [7,9]}			12	
X_{11}	{[1,4], [2,5], [3,10], [6,7], [8,9]}			6	
X_{12}	{[1,4], [2,5], [3,10], [6,8], [7,9]}			6	
X_{13}	{[1,4], [2,6], [3,7], [5,10], [8,9]}				2
X_{14}	{[1,4], [2,6], [3,8], [5,10], [7,9]}				1
X_{15}	{[1,4], [2,6], [3,9], [5,8], [7,10]}				3
X_{16}	{[1,4], [2,6], [3,10], [5,8], [7,9]}				1
X_{17}	{[1,4], [2,7], [3,5], [6,10], [8,9]}				3
X_{18}	{[1,4], [2,7], [3,6], [5,10], [8,9]}				2

FIGURE 3.3 (cont'd)

1-Factor	Edges [x,y]	M1	M2	M3	M4
X_{19}	{[1,4], [2,7], [3,9], [5,8], [6,10]}				1
X_{20}	{[1,4], [2,7], [3,9], [5,10], [6,8]}				1
X_{21}	{[1,4], [2,7], [3,10], [5,9], [6,8]}				2
X_{22}	{[1,4], [2,8], [3,5], [6,10], [7,9]}				2
X_{23}	{[1,4], [2,8], [3,6], [5,10], [7,9]}				1
X_{24}	{[1,4], [2,8], [3,9], [5,7], [6,10]}				1
X_{25}	{[1,4], [2,8], [3,9], [5,10], [6,7]}				1
X_{26}	{[1,4], [2,8], [3,10], [5,9], [6,7]}				2
X_{27}	{[1,4], [2,9], [3,5], [6,8], [7,10]}				1
X_{28}	{[1,4], [2,9], [3,6], [5,8], [7,10]}				1
X_{29}	{[1,4], [2,9], [3,7], [5,10], [6,8]}				1
X_{30}	{[1,4], [2,9], [3,8], [5,10], [6,7]}				1
X_{31}	{[1,4], [2,9], [3,10], [5,7], [6,8]}				1
X_{32}	{[1,4], [2,9], [3,10], [5,8], [6,7]}				1
X_{33}	{[1,4], [2,10], [3,5], [6,8], [7,9]}				1
X_{34}	{[1,4], [2,10], [3,6], [5,8], [7,9]}				1
X_{35}	{[1,4], [2,10], [3,9], [5,7], [6,8]}				1
X_{36}	{[1,4], [2,10], [3,9], [5,8], [6,7]}				1

LEMMA 3.7. *For* K_{10}, *let* W'_3 *be the set of 1-factors* X_j *in* W_3 *such that* $E(X_1 \cup X_2) \cap E(X_j) = \phi$, *where* X_1 *and* X_2 *are given in Figure 3.3*. *Then* W'_3 *has* 40 *members*. *By performing mappings which are automorphisms on both* X_1 *and* $X_1 \cup X_2$, *the set of* $X_1 \cup X_2 \cup X_j$ *for* $X_j \in W'_3$ *can be reduced to the* 7 *equivalence classes given in Figure 3.3 with entries in column M3*. *The numbers in column M3 give the number of members of* W'_3 *in each equivalence class*.

PROOF. By exhaustion using Computer Program 2. ■

LEMMA 3.8. *For* K_{10}, *let* W^2_3 *be the set of 1-factors* X_j *in* W_3 *such that* $E(X_1 \cup X_3) \cap E(X_j) = \phi$, *where* X_1 *and* X_3 *are given in Figure 3.3*. *Then* W^2_3 *has* 41 *members*. *By performing mappings which are automorphisms on* X_1 *and* $X_1 \cup X_3$ *the set of* $X_1 \cup X_3 \cup X_j$ *for* $X_j \in W^2_3$ *can be reduced to the* 28 *equivalence classes given in Figure 3.3 with entries in column M4*. *The numbers in column M4 give the number of members of* W^2_3 *in each equivalence class*.

PROOF. By exhaustion using Computer Program 2. ■

We shall now introduce some notation which we shall use throughout the rest of the chapter.

Let

$$F = \{Y_1, \ldots, Y_{2n-1}\}$$

represent a 1-factorization of K_{2n} such that each $Y_i \in W_i$. Then an alternating k-circuit in F is a circuit of length k such that the edges alternate between two 1-factors of F.

Let

$$A_k(F) = \{C : C \text{ is an alternating k-circuit for } F\}.$$

For $Y \in F$, let

$$A_k(F;Y) = \{C : C \text{ is an alternating k-circuit such that } E(C) \cap Y \neq \phi\}.$$

Let

$$P_k(F) = \{|A_k(F;Y)| : Y \in F\} .$$

For the vertex $x \in V(K_{2n})$ let

$$A_k(F;x) = \{C : C \text{ is an alternating k-circuit such that } x \in V(C)\} .$$

Let

$$Q_k(F) = \{|A_k(F;x)| : x \in V(K_{2n})\} .$$

The following lemma is useful for testing for isomorphism between two 1-factorizations of K_{2n}.

LEMMA 3.9. *Let* F *and* F' *be two equivalent 1-factorizations of* K_{2n} *and let* $\psi \in G(K_{2n})$ *such that* $\psi F = F'$. *Then*:

i. *For* $Y \in F$ *and* $Y' \in F'$, *if* $\psi Y = Y'$ *then* $|A_k(F;Y)| = |A_k(F';Y')|$ *for all* k .

ii. *For* $x, x' \in V(K_{2n})$, *if* $\psi x = x'$ *then* $|A_k(F;x)| = |A_k(F';x')|$ *for all* k .

iii. *For* $Y_i, Y_j \in F$ *and* $Y'_i, Y'_j \in F'$, *if* $\psi(Y_i \cup Y_j) = Y'_i \cup Y'_j$ *then* $\{|A_k(F;Y_i)|, |A_k(F;Y_j)|\} = \{|A_k(F',Y'_i)|, |A_k(F',Y'_j)|\}$ *for all* k .

PROOF. Follows immediately from the fact that isomorphism preserves adjacency and non-adjacency. ∎

The purpose of the next lemma is to enable us to determine the number of members in an equivalence class $L(F)$ without generating all the members of $L(F)$.

LEMMA 3.10. *Let* F *be defined as before and let*

$M_{k,i}(F) = \{F': F' \varepsilon L(F)$ and $|A_k(F;Y_1)| = |A_k(F';Y_i')|\}$ *for* $i = 1, \ldots, 2n-1$.

Then

$$|M_{k,1}(F)| = \cdots = |M_{k,2n-1}(F)| = M_k .$$

Also,

$$|L(F)| = \frac{M_k \cdot (2n-1)}{c}$$

where c $(c > 0)$ *is the multiplicity of* $|A_k(F,Y_1)|$ in $P_k(F)$.

PROOF. By Lemma 3.9 every $F' \varepsilon L(F)$ is a member of some $M_{k,i}(F)$. Since by Theorem 2.13 all 1-factors of K_{2n} are isomorphic, it is obvious that

$$|M_{k,1}(F)| = \cdots = |M_{k,2n-1}(F)|.$$

Let $M_k = |M_{k,1}(F)|$. If $|A_k(F,Y_1)|$ has multiplicity c in $P_k(F)$ then any $F' \varepsilon L(F)$ must be a member of c sets of $M_{k,i}(F)$ for $i = 1, \ldots, 2n-1$.

Therefore

$$|L(F)| = \frac{M_k \cdot (2n-1)}{c} . \quad ▮$$

LEMMA 3.11. *For any 1-factorization* F *of* K_{2n} *let* $A_k(F;Y_i)$ *and* $P_k(F)$ *be defined as above. Then at least one* $|A_k(F;Y_i)| \varepsilon P_k(F)$ *is even.*

PROOF. Let $|A_k(F)|$ represent the total number of alternating k-circuits in F. But every alternating circuit involves two 1-factors. Therefore, by the definition of $A_k(F;Y_i)$

$$\sum_{i=1}^{2n-1} |A_k(F;Y_i)| = 2\,|A_k(F)| \,.$$

Assume that every $|A_k(F;Y_i)| \varepsilon P_k(F)$ is odd.

Then since (2n-1) is odd $\sum_{i=1}^{2n-1} |A_k(F;Y_i)|$ must also be odd, which is a contradiction. Therefore at least one $|A_k(F;Y_i)| \varepsilon A_k(F)$ is even. ■

The following lemma enables us to subdivide the complete set of equivalence classes of 1-factorizations of K_{2n} into n subsets.

LEMMA 3.12. *Let* Φ *represent the complete set of 1-factorizations of* K_{2n}. *Then* Φ *can be partitioned into* n *subsets as follows:*

$\Phi(k,0) = \{F \in \Phi :$ there exists a $F_j \in L(F)$ such that $|A_k(F_j;Y_1)| = 0\}$.

For $\mu = 1, \ldots, n-1$,

$\Phi(k,\mu) = \{F \in \Phi :$ there exists a $F_j \in L(F)$ such that $|A_k(F_j;Y_1)| = 2\mu$ and

$\frac{1}{2}|A_k(F_j;Y_i)| \notin \{0, \ldots, \mu-1\}$ for all $i = 2, \ldots, 2n-1\}$.

PROOF. It is clear from the definition of $\Phi(k,\mu)$ for $\mu = 0, \ldots, n-1$ that if $F \in \Phi(k,\mu)$ then $L(F) \subseteq \Phi(k,\mu)$. By Lemma 3.11 we know that for every $F \in \Phi$ and every even k such that $4 \leq k \leq 2n$, some $|A_k(F;Y_i)| = 2\mu$ for some $\mu \in \{0,1, \ldots, n-1\}$. By Lemma 3.10 we know that if $|A_k(F;Y_i)| = 2\mu$ for some $\mu \in \{0,1, \ldots, n-1\}$ there exists a $F_j \in L(F)$ such that $|A_k(F_j;Y_1)| = 2\mu$. Clearly, therefore, $\Phi = \Phi(k,0) \cup \Phi(k,1) \cup \ldots \cup \Phi(k,n-1)$. But, for $\Phi(k,\gamma)$ and $\Phi(k,\delta)$ such that $\gamma, \delta \in \{0,1, \ldots, n-1\}$ and $\delta > \gamma$ we have $2\gamma \in P_k(F_j)$ for every $F_j \in \Phi(k,\gamma)$ whereas $2\gamma \notin P_k(F_m)$ for every $F_m \in \Phi(k,\delta)$. Therefore, $\Phi(k,\gamma) \cap \Phi(k,\delta) = \phi$ for every $\gamma, \delta \in \{0,1, \ldots, n-1\}$ and $\gamma \neq \delta$. ∎

THEOREM 3.13. <u>There are</u> 396 <u>equivalence classes of 1-factorizations of</u> K_{10}.

PROOF. In our procedure for enumerating equivalence classes of 1-factorizations of K_{10}, we considered alternating Hamiltonian circuits on the 1-factors and alternating 4-circuits on the vertices. We partitioned the set of equivalence classes into five subsets, $\Phi(10,0)$, ..., $\Phi(10,4)$, based on Lemma 3.12. Using Lemmas 3.6, 3.7 and 3.8 we only considered schedules which included the 1-factors listed in Figure 3.3. Computer Program 2 followed the backtracking technique described in Figure 3.1 but restricted itself to only produce schedules such that X_1 formed 2μ Hamiltonian circuits, where the value 2μ for $\mu \varepsilon \{0, \ldots, 4\}$ was read into the program from a punched card.

As a schedule was generated by the computer program, it was tested to see if it represented a new equivalence class. Let F represent the current schedule being tested and let F' represent a previously-generated representative of an equivalence class. By Lemma 3.9 the program performed the following tests for isomorphism:

1) $|A_{10}(F)| = |A_{10}(F')|$.

2) $P_{10}(F) = P_{10}(F')$ and $Q_4(F) = Q_4(F')$.

If F and F' matched on these tests then the computer program performed a test based on part iii of Lemma 3.9. That is, it performed mappings on alternating circuits of F and F' in an order based on numbers of low multiplicity in $P_{10}(F)$ and $Q_4(F)$. If a mapping ψ was found such that $\psi F = F'$ then the count for the equivalence class was incremented by one. If no mapping could be found, the computer program created a new equivalence class with F as its representative. ∎

See Appendix A for a listing of representatives of each equivalence class of 1-factorizations of K_{10}. Included with each schedule F are the sets $P_{10}(F)$ and $Q_4(F)$ as well as the number of Hamiltonian circuits, $|A_{10}(F)|$.

THEOREM 3.14. <u>The order of the automorphism group of each of the</u> 396 <u>equivalence classes of 1-factorizations of</u> K_{10} <u>is as given in Appendix A</u>.

PROOF. By Theorem 3.13, Computer Program 2 generated and counted all schedules, F, in each equivalence class which have the following characteristics:

1) $X_1 \in F$.

2) X_2 or X_3 is in F where X_2, X_3 are given in Figure 3.3.

3) If $X_2 \in F$ then one of $X_5, X_7, X_8, X_9, X_{10}, X_{11}$ or X_{12} listed in Figure 3.3 is in F. If $X_3 \in F$ then one of $X_4, \ldots, X_7, X_{13}, \ldots, X_{36}$ is in F.

4) For $\mu = 0, \ldots, n-1$, $F \in \Phi(10,\mu)$ and $|A_{10}(F;X_1)| = 2\mu$. (See Lemma 3.12).

By Lemmas 3.6, 3.7, 3.8, each total generated by the computer program can be multiplied by the multiplication factors M1, M2 and M3 or M4 from Figure 3.3. By Lemma 3.10, each total can then be multiplied by 9 and divided by c (where c is the multiplicity of 2μ in $P_{10}(F)$) to give the total number of members in each equivalence class, $|L(F)|$.

Finally, by Theorem 2.16, $|G(F)| = \dfrac{10!}{|L(F)|}$. ∎

THEOREM 3.15. <u>There are</u> 1,225,566,720 <u>1-factorizations of</u> K_{10} <u>labelled on the set</u> $\{1,2, \ldots, 10\}$.

PROOF. $\sum_{i=1}^{396} |L(F_i)| = 1{,}225{,}566{,}720$ where the 1-factorizations, F_i are listed in Appendix A and each $|L(F_i)|$ was found by Computer Program 2 by Theorem 3.14. ∎

As noted earlier, listed with the representatives of the equivalence classes of 1-factorizations of K_{10} in Appendix A are the sets of numbers $P_{10}(F_i)$, (the set of counts of alternating Hamiltonian circuits across the 1-factors), and $Q_4(F_i)$, (the set of counts of alternating 4-circuits across the vertices). Unfortunately, these numbers do not completely characterize the equivalence classes. There are six pairs of equivalence classes with the same two sets of numbers (see Appendix A): $\{F_{85}, F_{86}), (F_{176}, F_{177}), (F_{223}, F_{224}), (F_{267}, F_{268}), (F_{290}, F_{291}), (F_{303}, F_{304})\}$, and other tests must be performed in order to differentiate between them. We were unable to find an easily calculable method of completely characterizing the 396 equivalence classes of 1-factorizations of K_{10}.

It is of interest to note that the 1-factorization of K_{10} given by Theorem 2.6 is in the equivalence class $L(F_{290})$ where F_{290} is given in Appendix A. It has 27 Hamiltonian circuits and is therefore (as noted in Corollary 2.12) not an H-schedule. The order of its automorphism group, $|G(F_{290})|$, is 54.

We note also that K_{10} is the smallest complete graph which has a non-symmetrical 1-factorization (i.e., a 1-factorization which does not have any non-trivial automorphisms). In fact, of the 396 equivalence classes of 1-factorizations of K_{10}, 298 of them are non-symmetrical.

3-4. Conclusion

In our study we have counted the number and size of the equivalence classes of 1-factorizations of K_{2n} for $n \leq 5$. For larger n, the problem appears much more difficult, simply because of the large number of 1-factorizations involved. However, it should be possible to count the equivalence classes of 1-factorizations of K_{12} by a generalization of Lemmas 3.6, 3.7, and 3.8, and extending the method through each subset W_i (see Sections 3-1 and 3-3 of Chapter 3) of the 1-factors of K_{12}. In other words, by constructively creating schedules round by round, testing for isomorphism at each stage, it should be possible to produce a complete set of non-equivalent schedules with 12 teams in a fairly "reasonable" length of time.

It seems clear, however, that the number of equivalence classes of 1-factorizations (as well as the total number of 1-factorizations) of K_{2n} increases very rapidly as n increases. Therefore, to solve the problem for 2n larger than 12 by the methods of this thesis would appear to be extremely difficult, if not impossible.

CHAPTER 4

AN APPLICATION

Round robin schedules are commonly used for sporting matches and tournaments. Because of its ease of generation, the typical round robin schedule chosen is the scheme given by Theorem 2.6. We will show that, when given certain constraints, the choice of the round robin schedule can spell the difference between success or failure in solving the particular problem.

The problem is to create a schedule for sports such as curling, tennis, bowling, etc., where the number of playing fields are at a premium. The schedule must be as equitable as possible; that is, no team will have an advantage by playing at a particular favourable site more often than other teams. The schedule will be a round robin schedule involving $2n$ teams. Because each round will contain n games the number of fields to play on will also be n. In order to maximize the fairness of the schedule, the following two constraints are to be satisfied:

(i) no team plays more than twice on the same field,

(ii) no team plays twice in a row on the same field.

We chose the situation where we had 8 teams playing a round robin and 4 fields at our disposal. By Theorem 3.3

there are 6 equivalence classes of schedules with 8 teams (listed in Figure 3.2). A computer program was written to use a representative schedule from each of the six equivalence classes and to place games on fields round by round in an attempt to pass constraint (i). If it was successful it then reordered the rounds in an attempt to pass constraint (ii).

It was found that $L(F_2)$, $L(F_4)$ and $L(F_5)$ were the only round robin schedules which could pass the two constraints. The schedule $L(F_6)$ given by Theorem 2.6 passed constraint (i) but could not pass constraint (ii). The remaining two equivalence classes, $L(F_1)$ and $L(F_3)$, could not pass constraint (i).

Figure 4.1 below gives a round robin schedule with 8 teams and 4 fields which will pass the constraints. The schedule chosen is F_4 from Figure 3.2.

FIGURE 4.1 A ROUND ROBIN SCHEDULE WITH 8 TEAMS AND 4 FIELDS.

Round	Games	Field	Round	Games	Field
1	1 - 2	1	2	1 - 5	4
	3 - 4	2		2 - 6	2
	5 - 6	3		3 - 8	3
	7 - 8	4		4 - 7	1

FIGURE 4.1 (cont'd)

Round	Games	Field	Round	Games	Field
3	1 - 3	1	4	1 - 4	2
	2 - 4	3		2 - 3	4
	5 - 7	2		5 - 8	1
	6 - 8	4		6 - 7	3
5	1 - 7	4	6	1 - 8	3
	2 - 8	2		2 - 5	1
	3 - 6	1		3 - 7	2
	4 - 5	3		4 - 6	4
7	1 - 6	2			
	2 - 7	3			
	3 - 5	4			
	4 - 8	1			

Note: Game $x - y$ corresponds to team x playing team y.

We note that the schedule is cyclic with respect to the rounds; i.e., the rounds can be played in the order $i \pmod 7$, $i+1 \pmod 7$, ..., $i+6 \pmod 7$. Clearly they can also be played in the reverse order.

BIBLIOGRAPHY

1. ANDERSON, S. S. (1970). Graph theory and finite combinations. Markham, Chicago.

2. BEHZAD, M. and G. CHARTRAND (1971). Introduction to the theory of graphs. Allyn and Bacon, Boston.

3. BEINEKE, L. W. and M. D. PLUMMER (1967). On the 1-factors of a non-separable graph. J. Comb. Theory, Vol. 2, pp. 285-289.

4. BUSACKER, R. G. and T. L. SAATY (1965). Finite Graphs and Networks. McGraw-Hill, New York.

5. CORNEIL, D. G. and C. C. GOTLIEB (1970). An efficient algorithm for graph isomorphism. J. for A.C.M., Vol. 17, pp. 51-64.

6. FREUND, J. E. (1956). Round robin mathematics. Amer. Math.Monthly, Vol. 63, pp. 112-114.

7. GOLOMB, S. W. and L. D. BAUMERT (1965). Backtrack Programming. J. for A.C.M., Vol. 12, pp. 516-524.

8. HARARY, F. (1969). Graph Theory. Addison-Wesley, Reading, Mass.

9. HARARY, F. (ed.) (1969). Proof techniques in graph theory. Academic Press, New York.

10. KÖNIG, D. (1936). Theorie der endlichen und unendlichen graphen. Chelsea, New York.

11. KRAITCHIK, M. (1953). Mathematical Recreations. Dover, New York.

12. LOCKWOOD, E. H. (1936). American tournament. Math. Gazette, Vol. 20, p. 333.

13. LOCKWOOD, E. H. (1962). Tournament problems. Math. Gazette, Vol. 46, p. 220.

14. MARSHALL, C. W. (1971). Applied graph theory. Wiley-Interscience, New York.

15. MAXWELL, L. M. and M. B. REED (1971). The theory of graphs: a basis for Network Theory. Pergamon, New York.

16. MOON, J. W. (1968). Topics on Tournaments. Holt, Rinehart and Winston, New York.

17. ORE, O. (1963). Graphs and their uses. Random House, New York.

18. POLLACK, S. V. and T. D. STERLING (1969). A Guide to PL/1. Holt, Rinehart and Winston, New York.

19. REISZ, M. (1859). Über eine Steinersche kombinatorische Aufgabe welche in 45 sten Bande dieses Journals, Seite 181, gestellt worden ist. J. Reine. Angew. Math., Vol. 56, pp. 326-344.

20. SHEEHAN, J. (1968). On the number of graphs with a given automorphism group. Theory of Graphs: Proceedings, edited by P. Erdos and G. Katona. Academic Press, New York.

21. TUTTE, W. T. (1947). The factorizations of linear graphs. J. London Math. Soc., Vol. 22, pp. 107-111.

22. TUTTE, W. T. (1966). Connectivity in Graphs. University of Toronto, Toronto.

23. WALKER, R. L. (1960). An emmerative technique for a class of combinatorial problems. Amer. Math. Soc. Proc. Symp. Appl. Math., Vol. 10, pp. 91-94.

24. WILSON, R. J. (1972). Introduction to Graph Theory. Oliver and Boyd, Edinburgh.

25. ZAKS, J. Y. (1971). On the 1-factors of n-connected graphs. J. Comb. Theory (B), Vol. 11, pp. 169-180.

APPENDIX A

Representatives from each of the equivalence classes of 1-factorizations of K_{10}.

Appendix A consists of 3 listings:

1. Factor list - a listing of the 1-factors X_i ($i = 1, \ldots, 270$) which are contained in the subsequent schedules.
2. Schedules by factor number - a listing of the 396 representative schedules. The numbers under the columns labelled "FACTORS" refer to the 1-factors in the Factor list. The "ORDER" is the order of the automorphism group. The column "HC" gives the total number of alternating Hamiltonian circuits in the schedule. The columns labelled "# HAMILT CIRCUITS ACROSS THE FCTRS" gives the numbers in $P_{10}(F_i)$ ordered by factor number. The columns labelled "# 4-CIRCUITS ACROSS THE VERTICES" give the numbers in $Q_4(F_i)$ ordered by vertex number.
3. Auxiliary listing to the list of schedules - The numbers in $P_{10}(F_i)$ and $Q_4(F_i)$ have been ordered from smallest to largest. This listing gives a partial characterization of the 396 equivalence classes.

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

FACTOR LIST

FCTR     EDGES                                      FCTR     EDGES

  1      1- 2   3- 4   5- 6   7- 8   9-10             2      1- 3   2- 4   5- 7   6- 9   8-10
  3      1- 3   2- 5   4- 7   6- 9   8-10             4      1- 4   2- 3   5- 8   6-10   7- 9
  5      1- 4   2- 3   5-10   6- 8   7- 9             6      1- 4   2- 5   3- 6   7-10   8- 9
  7      1- 4   2- 5   3- 7   6-10   8- 9             8      1- 4   2- 5   3- 8   6-10   7- 9
  9      1- 4   2- 5   3-10   6- 7   8- 9            10      1- 4   2- 6   3- 7   5-10   8- 9
 11      1- 4   2- 6   3- 8   5-10   7- 9            12      1- 4   2- 6   3- 9   5- 8   7-10
 13      1- 4   2- 6   3-10   5- 8   7- 9            14      1- 5   2- 3   4- 8   6-10   7- 9
 15      1- 5   2- 3   4- 9   6- 8   7-10            16      1- 5   2- 3   4-10   6- 7   8- 9
 17      1- 5   2- 3   4-10   6- 8   7- 9            18      1- 5   2- 4   3- 9   6- 8   7-10
 19      1- 5   2- 4   3-10   6- 8   7- 9            20      1- 5   2- 6   3- 7   4-10   8- 9
 21      1- 5   2- 6   3- 8   4- 9   7-10            22      1- 5   2- 6   3- 8   4-10   7- 9
 23      1- 5   2- 5   3- 9   4- 8   7-10            24      1- 5   2- 6   3-10   4- 8   7- 9
 25      1- 5   2- 7   3- 6   4-10   8- 9            26      1- 5   2- 7   3- 8   4- 9   6-10
 27      1- 5   2- 7   3- 9   4- 8   6-10            28      1- 5   2- 7   3- 9   4-10   6- 8
 29      1- 5   2- 7   3-10   4- 6   8- 9            30      1- 5   2- 7   3-10   4- 9   6- 8
 31      1- 5   2- 8   3- 6   4- 9   7-10            32      1- 5   2- 8   3- 6   4-10   7- 9
 33      1- 5   2- 8   3- 7   4- 9   6-10            34      1- 5   2- 8   3- 9   4- 6   7-10
 35      1- 5   2- 8   3- 9   4-10   6- 7            36      1- 5   2- 8   3-10   4- 6   7- 9
 37      1- 5   2- 8   3-10   4- 9   6- 7            38      1- 5   2- 9   3- 6   4- 8   7-10
 39      1- 5   2- 9   3- 7   4-10   6- 8            40      1- 5   2- 9   3- 8   4-10   6- 7
 41      1- 5   2- 9   3-10   4- 7   6- 8            42      1- 5   2- 9   3-10   4- 8   6- 7
 43      1- 5   2-10   3- 6   4- 8   7- 9            44      1- 5   2-10   3- 7   4- 9   6- 8
 45      1- 6   2- 3   4- 8   5- 9   7-10            46      1- 6   2- 3   4- 8   5-10   7- 9
 47      1- 6   2- 3   4- 9   5- 8   7-10            48      1- 6   2- 3   4-10   5- 7   8- 9
 49      1- 6   2- 3   4-10   5- 8   7- 9            50      1- 6   2- 4   3- 8   5- 9   7-10
 51      1- 6   2- 4   3- 8   5-10   7- 9            52      1- 6   2- 4   3- 9   5- 8   7-10
 53      1- 6   2- 4   3-10   5- 7   8- 9            54      1- 6   2- 4   3-10   5- 8   7- 9
 55      1- 6   2- 5   3- 7   4-10   8- 9            56      1- 6   2- 5   3- 8   4- 9   7-10
 57      1- 6   2- 5   3- 9   4- 8   7-10            58      1- 6   2- 7   3- 5   4-10   8- 9
 59      1- 6   2- 7   3- 8   4- 9   5-10            60      1- 6   2- 7   3- 8   4-10   5- 9
 61      1- 6   2- 7   3- 9   4- 8   5-10            62      1- 6   2- 7   3- 9   4-10   5- 8
 63      1- 6   2- 7   3-10   4- 5   8- 9            64      1- 6   2- 7   3-10   4- 8   5- 9
 65      1- 6   2- 7   3-10   4- 9   5- 8            66      1- 6   2- 8   3- 5   4- 9   7-10
 67      1- 6   2- 8   3- 5   4-10   7- 9            68      1- 6   2- 8   3- 7   4- 9   5-10
 69      1- 6   2- 8   3- 7   4-10   5- 9            70      1- 6   2- 8   3- 9   4- 5   7-10
 71      1- 6   2- 8   3- 9   4- 7   5-10            72      1- 6   2- 8   3-10   4- 5   7- 9
 73      1- 6   2- 8   3-10   4- 7   5- 9            74      1- 6   2- 9   3- 5   4- 8   7-10
 75      1- 6   2- 9   3- 7   4- 8   5-10            76      1- 6   2- 9   3- 7   4-10   5- 8
 77      1- 6   2- 9   3- 8   4- 5   7-10            78      1- 6   2- 9   3- 8   4- 7   5-10
 79      1- 6   2- 9   3- 8   4-10   5- 7            80      1- 6   2- 9   3-10   4- 7   5- 8
 81      1- 6   2- 9   3-10   4- 8   5- 7            82      1- 6   2-10   3- 5   4- 7   8- 9
 83      1- 6   2-10   3- 5   4- 8   7- 9            84      1- 6   2-10   3- 7   4- 5   8- 9
 85      1- 6   2-10   3- 7   4- 8   5- 9            86      1- 6   2-10   3- 7   4- 9   5- 8
 87      1- 6   2-10   3- 8   4- 5   7- 9            88      1- 6   2-10   3- 8   4- 7   5- 9
 89      1- 6   2-10   3- 8   4- 9   5- 7            90      1- 6   2-10   3- 9   4- 7   5- 8
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

FACTOR LIST

FCTR      EDGES                                          FCTR      EDGES

  91    1- 6   2-10   3- 9   4- 8   5- 7                   92    1- 7   2- 3   4- 5   6-10   8- 9
  93    1- 7   2- 3   4- 6   5-10   8- 9                   94    1- 7   2- 3   4- 8   5- 9   6-10
  95    1- 7   2- 3   4- 9   5- 8   6-10                   96    1- 7   2- 3   4- 9   5-10   6- 8
  97    1- 7   2- 3   4-10   5- 9   6- 8                   98    1- 7   2- 4   3- 5   6-10   8- 9
  99    1- 7   2- 4   3- 9   5- 8   6-10                  100    1- 7   2- 4   3-10   5- 9   6- 8
 101    1- 7   2- 5   3- 6   4-10   8- 9                  102    1- 7   2- 5   3- 8   4- 9   6-10
 103    1- 7   2- 5   3- 9   4- 8   6-10                  104    1- 7   2- 5   3-10   4- 9   6- 8
 105    1- 7   2- 6   3- 5   4-10   8- 9                  106    1- 7   2- 6   3- 8   4- 9   5-10
 107    1- 7   2- 6   3- 8   4-10   5- 9                  108    1- 7   2- 6   3- 9   4- 8   5-10
 109    1- 7   2- 6   3- 9   4-10   5- 8                  110    1- 7   2- 6   3-10   4- 5   8- 9
 111    1- 7   2- 6   3-10   4- 8   5- 9                  112    1- 7   2- 6   3-10   4- 9   5- 8
 113    1- 7   2- 8   3- 5   4- 9   6-10                  114    1- 7   2- 8   3- 6   4- 9   5-10
 115    1- 7   2- 8   3- 9   4- 5   6-10                  116    1- 7   2- 8   3- 9   4- 6   5-10
 117    1- 7   2- 8   3-10   4- 6   5- 9                  118    1- 7   2- 9   3- 5   4- 8   6-10
 119    1- 7   2- 9   3- 5   4-10   6- 8                  120    1- 7   2- 9   3- 6   4- 8   5-10
 121    1- 7   2- 9   3- 6   4-10   5- 8                  122    1- 7   2- 9   3- 8   4- 5   6-10
 123    1- 7   2- 9   3- 8   4- 6   5-10                  124    1- 7   2- 9   3-10   4- 5   6- 8
 125    1- 7   2- 9   3-10   4- 6   5- 8                  126    1- 7   2-10   3- 5   4- 9   6- 8
 127    1- 7   2-10   3- 6   4- 8   5- 9                  128    1- 7   2-10   3- 6   4- 9   5- 8
 129    1- 7   2-10   3- 8   4- 6   5- 9                  130    1- 7   2-10   3- 9   4- 5   6- 8
 131    1- 7   2-10   3- 9   4- 6   5- 8                  132    1- 8   2- 3   4- 5   6-10   7- 9
 133    1- 8   2- 3   4- 6   5- 9   7-10                  134    1- 8   2- 3   4- 6   5-10   7- 9
 135    1- 8   2- 3   4- 7   5- 9   6-10                  136    1- 8   2- 3   4- 9   5- 7   6-10
 137    1- 8   2- 3   4- 9   5-10   6- 7                  138    1- 8   2- 3   4-10   5- 9   6- 7
 139    1- 8   2- 4   3- 5   6-10   7- 9                  140    1- 8   2- 4   3- 6   5- 9   7-10
 141    1- 8   2- 4   3- 9   5- 7   6-10                  142    1- 8   2- 4   3-10   5- 9   6- 7
 143    1- 8   2- 5   3- 6   4- 9   7-10                  144    1- 8   2- 5   3- 7   4- 9   6-10
 145    1- 8   2- 5   3- 9   4- 6   7-10                  146    1- 8   2- 5   3- 9   4- 7   6-10
 147    1- 8   2- 5   3-10   4- 9   6- 7                  148    1- 8   2- 6   3- 5   4- 9   7-10
 149    1- 8   2- 6   3- 5   4-10   7- 9                  150    1- 8   2- 6   3- 7   4- 9   5-10
 151    1- 8   2- 6   3- 7   4-10   5- 9                  152    1- 8   2- 6   3- 9   4- 7   5-10
 153    1- 8   2- 6   3-10   4- 5   7- 9                  154    1- 8   2- 6   3-10   4- 7   5- 9
 155    1- 8   2- 7   3- 5   4- 9   6-10                  156    1- 8   2- 7   3- 6   4- 9   5-10
 157    1- 8   2- 7   3- 6   4-10   5- 9                  158    1- 8   2- 7   3- 9   4- 5   6-10
 159    1- 8   2- 7   3- 9   4- 6   5-10                  160    1- 8   2- 7   3-10   4- 6   5- 9
 161    1- 8   2- 9   3- 5   4- 6   7-10                  162    1- 8   2- 9   3- 5   4- 7   6-10
 163    1- 8   2- 9   3- 5   4-10   6- 7                  164    1- 8   2- 9   3- 6   4- 5   7-10
 165    1- 8   2- 9   3- 6   4- 7   5-10                  166    1- 8   2- 9   3- 6   4-10   5- 7
 167    1- 8   2- 9   3- 7   4- 5   6-10                  168    1- 8   2- 9   3- 7   4- 6   5-10
 169    1- 8   2- 9   3-10   4- 5   6- 7                  170    1- 8   2- 9   3-10   4- 6   5- 7
 171    1- 8   2-10   3- 5   4- 6   7- 9                  172    1- 8   2-10   3- 5   4- 9   6- 7
 173    1- 8   2-10   3- 6   4- 5   7- 9                  174    1- 8   2-10   3- 6   4- 7   5- 9
 175    1- 8   2-10   3- 6   4- 9   5- 7                  176    1- 8   2-10   3- 7   4- 6   5- 9
 177    1- 8   2-10   3- 9   4- 5   6- 7                  178    1- 8   2-10   3- 9   4- 6   5- 7
 179    1- 9   2- 3   4- 5   6- 8   7-10                  180    1- 9   2- 3   4- 6   5- 8   7-10
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

FACTOR LIST

FCTR      EDGES                                     .FCTR      EDGES

181    1- 9   2- 3   4- 7   5- 8   6-10            182    1- 9   2- 3   4- 7   5-10   6- 8
183    1- 9   2- 3   4- 8   5- 7   6-10            184    1- 9   2- 3   4- 8   5-10   6- 7
185    1- 9   2- 4   3- 5   6- 8   7-10            186    1- 9   2- 4   3- 6   5- 8   7-10
187    1- 9   2- 4   3- 7   5- 8   6-10            188    1- 9   2- 4   3- 7   5-10   6- 8
189    1- 9   2- 4   3- 8   5- 7   6-10            190    1- 9   2- 4   3- 8   5-10   6- 7
191    1- 9   2- 5   3- 6   4- 8   7-10            192    1- 9   2- 5   3- 8   4- 6   7-10
193    1- 9   2- 5   3- 8   4- 7   6-10            194    1- 9   2- 5   3-10   4- 7   6- 8
195    1- 9   2- 5   3-10   4- 8   6- 7            196    1- 9   2- 6   3- 5   4- 8   7-10
197    1- 9   2- 6   3- 7   4- 8   5-10            198    1- 9   2- 6   3- 7   4-10   5- 8
199    1- 9   2- 6   3- 8   4- 5   7-10            200    1- 9   2- 6   3- 8   4- 7   5-10
201    1- 9   2- 6   3-10   4- 7   5- 8            202    1- 9   2- 7   3- 5   4- 8   6-10
203    1- 9   2- 7   3- 5   4-10   6- 8            204    1- 9   2- 7   3- 6   4- 8   5-10
205    1- 9   2- 7   3- 6   4-10   5- 8            206    1- 9   2- 7   3- 8   4- 5   6-10
207    1- 9   2- 7   3- 8   4- 6   5-10            208    1- 9   2- 7   3-10   4- 5   6- 8
209    1- 9   2- 7   3-10   4- 6   5- 8            210    1- 9   2- 8   3- 5   4- 6   7-10
211    1- 9   2- 8   3- 5   4- 7   6-10            212    1- 9   2- 8   3- 5   4-10   6- 7
213    1- 9   2- 8   3- 6   4- 5   7-10            214    1- 9   2- 8   3- 6   4- 7   5-10
215    1- 9   2- 8   3- 6   4-10   5- 7            216    1- 9   2- 8   3- 7   4- 5   6-10
217    1- 9   2- 8   3- 7   4- 6   5-10            218    1- 9   2- 8   3-10   4- 5   6- 7
219    1- 9   2- 8   3-10   4- 6   5- 7            220    1- 9   2-10   3- 5   4- 7   6- 8
221    1- 9   2-10   3- 5   4- 8   6- 7            222    1- 9   2-10   3- 6   4- 7   5- 8
223    1- 9   2-10   3- 7   4- 5   6- 8            224    1- 9   2-10   3- 7   4- 6   5- 8
225    1- 9   2-10   3- 8   4- 5   6- 7            226    1-10   2- 3   4- 5   6- 7   8- 9
227    1-10   2- 3   4- 5   6- 8   7- 9            228    1-10   2- 3   4- 6   5- 7   8- 9
229    1-10   2- 3   4- 6   5- 8   7- 9            230    1-10   2- 3   4- 7   5- 9   6- 8
231    1-10   2- 3   4- 8   5- 9   6- 7            232    1-10   2- 3   4- 9   5- 8   6- 7
233    1-10   2- 4   3- 5   6- 7   8- 9            234    1-10   2- 4   3- 5   6- 8   7- 9
235    1-10   2- 4   3- 6   5- 7   8- 9            236    1-10   2- 4   3- 6   5- 8   7- 9
237    1-10   2- 4   3- 7   5- 9   6- 8            238    1-10   2- 4   3- 8   5- 9   6- 7
239    1-10   2- 5   3- 6   4- 7   8- 9            240    1-10   2- 5   3- 7   4- 6   8- 9
241    1-10   2- 5   3- 7   4- 9   6- 8            242    1-10   2- 5   3- 8   4- 9   6- 7
243    1-10   2- 5   3- 9   4- 7   6- 8            244    1-10   2- 5   3- 9   4- 8   6- 7
245    1-10   2- 6   3- 5   4- 8   7- 9            246    1-10   2- 6   3- 7   4- 5   8- 9
247    1-10   2- 6   3- 7   4- 8   5- 9            248    1-10   2- 6   3- 7   4- 9   5- 8
249    1-10   2- 6   3- 8   4- 5   7- 9            250    1-10   2- 6   3- 8   4- 7   5- 9
251    1-10   2- 6   3- 9   4- 7   5- 8            252    1-10   2- 7   3- 5   4- 9   6- 8
253    1-10   2- 7   3- 6   4- 8   5- 9            254    1-10   2- 7   3- 6   4- 9   5- 8
255    1-10   2- 7   3- 8   4- 6   5- 9            256    1-10   2- 7   3- 9   4- 5   6- 8
257    1-10   2- 7   3- 9   4- 6   5- 8            258    1-10   2- 8   3- 5   4- 6   7- 9
259    1-10   2- 8   3- 5   4- 9   6- 7            260    1-10   2- 8   3- 6   4- 7   5- 9
261    1-10   2- 8   3- 6   4- 9   5- 7            262    1-10   2- 8   3- 7   4- 6   5- 9
263    1-10   2- 8   3- 9   4- 5   6- 7            264    1-10   2- 8   3- 9   4- 6   5- 7
265    1-10   2- 9   3- 5   4- 7   6- 8            266    1-10   2- 9   3- 5   4- 8   6- 7
267    1-10   2- 9   3- 6   4- 7   5- 8            268    1-10   2- 9   3- 7   4- 5   6- 8
269    1-10   2- 9   3- 7   4- 6   5- 8            270    1-10   2- 9   3- 8   4- 5   6- 7
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES BY FACTOR NUMBER

SCHED    *   *   *   *   FACTORS   *   *   *   *  ORDER  HC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

  1      1   2   5  20  57 113 160 222 270   432   0     0  0  0  0  0  0  0  0  0    36 12 12 12 12 12 12 12 12 12
  2      1   2   5  20  57 113 160 225 267    12  12     0  2  2  2  2  2  2  6  6    24 12  8  8  8  8  8  8  6  6
  3      1   2   4  20  57 114 160 220 270     6  12     0  2  4  3  3  4  2  4  2    24 12  8  8  6  6  8  8  8  8
  4      1   2   4  20  57 114 160 225 265     2  16     0  3  5  3  3  5  3  6  4    18 14  7  5  6  6  5  7  8  4
  5      1   2   5  20  57 117 155 225 267    16  16     0  4  4  4  4  4  4  4  4    16 16  6  6  6  6  6  6  6  6
  6      1   2   4  20  57 129 156 218 265     2  18     2  4  6  2  2  6  4  6  4    10 16  8  6  7  7  5  4  5  4
  7      1   2   4  20  57 114 169 220 255     6  18     2  2  2  5  5  5  5  5  5    18  8  8  8  5  5  5  5  5  5
  8      1   2   5  20  65 127 146 210 270     4  18     2  2  4  4  5  5  5  5  4    12  7  7 14  8  4  4  4  4  6
  9      1   2   3  20  64 113 145 225 267     4  18     2  4  2  4  5  4  5  5  5    18  8  8  8  5  5  5  5  5  5
 10      1   2   4  28  75 117 148 225 239     2  18     2  4  4  4  4  5  4  5  4     5 14 12  7  4  4  8  5  8  5
 11      1   2   5  25  80 101 177 196 262    18  18     6  6  6  3  3  3  3  3  3     6 18  6  6  6  6  6  6  6  6
 12      1   2   5  23  55 125 172 206 260    12  18     4  4  4  4  4  4  4  4  4     8 18  5  5  8  8  5  5  5  5
 13      1   2   5  23  63 102 176 212 267     6  18     4  4  4  4  4  4  4  4  4    10 10 12 10  5  5  5  5  5  5
 14      1   2   4  20  61 104 174 210 270     1  19     2  3  2  4  6  5  6  6  4    11  9 11  8  7  4  3  3  7  5
 15      1   2   4  20  61 117 143 220 270     1  19     2  3  4  5  4  6  6  5  3    13 11  8  6  6  2  5  2  8  7
 16      1   2   4  20  61 129 147 213 265     1  20     2  2  5  4  5  7  5  4  6    10  8 12  6  3  8  3  2  7  5
 17      1   2   4  20  77 117 156 221 243     2  20     2  3  0  5  6  6  3  3  6    11 11  7  8  4  4  7  8  2  2
 18      1   2   5  20  65 103 174 210 270     2  20     2  4  4  3  6  6  6  6  3    14  5  7 10  7  4  3  3  4  7
 19      1   2   5  20  57 117 162 225 254     2  20     2  4  4  3  3  6  6  6  6    12 12  8  6  6  0  4  3  4  3
 20      1   2   4  20  64 114 145 220 270     1  20     2  4  4  5  7  4  6  4  4    15  8  6  9  3  3  5  3  7  5
 21      1   2   4  20  57 124 174 207 259     2  20     4  4  4  3  3  6  5  6  5    12 12  7  3  7  7  4  4  4  4
 22      1   3  10  14  50 124 172 215 257     8  20     8  4  4  4  4  4  4  4  4     7  7  7  7  7  7  7  7  8  C
 23      1   2   4  20  57 129 165 218 252     2  20     4  4  6  4  4  4  5  4  5    10 14  6  6  6  6  4  4  4  4
 24      1   2   4  25  74 117 150 225 243     2  20     4  4  6  4  4  5  4  5  4     6 10 12  8  5  5  5  4  5  4
 25      1   3  10  14  52 126 165 215 255     1  21     8  2  6  5  4  6  3  4  4    14  4  9  3  4  5  4  7  6  4
 26      1   2   4  20  56 116 169 220 253     1  21     2  4  3  4  4  7  5  6  7    12 11  7  6  7  4  4  4  3  2
 27      1   2   4  20  78 117 143 221 256     1  21     2  3  5  5  5  6  6  5  5    11  9  7  9  4  3  6  5  3  3
 28      1   2   6  17  78 112 158 221 262     1  21     2  4  5  7  6  4  4  4  6     5  9  6  7  3  2  7  7  7  7
 29      1   2   4  20  61 117 143 225 265     1  21     2  4  5  5  4  5  7  5  5    10 12  8  3  4  3  4  3  7  6
 30      1   2   4  25  77 116 147 220 247     1  21     2  5  5  6  6  5  4  4  5     7 11  9  9  2  2  5  5  5  5
 31      1   2   6  17  65 115 168 221 250     1  21     2  4  5  5  6  6  5  5  4     7  7  6  8  2  4  5  8  6  7
 32      1   2   7  15  62 127 165 200 258     1  21     4  3  3  6  3  7  6  5  5     7  4  8  8  3 10  2  6  3  7
 33      1   3  10  14  50 119 175 209 263     1  21     8  6  3  4  4  4  3  6  4    10  8  6  7  2  6  3  6  5  7
 34      1   2   5  20  66 125 177 193 253     1  21     4  3  4  4  6  6  5  3  7     8  9 10  9  6  3  5  3  5  2
 35      1   2   4  25  77 117 150 221 243     1  21     4  3  6  5  7  6  4  3  4     6  9  8 11  2  5  5  6  5  3
 36      1   2   5  25  70 118 176 201 242     6  21     6  3  3  5  5  5  5  5  5    15  3  6  6  5  5  5  5  5  5
 37      1   2   5  20  64 131 164 211 242     1  21     4  4  3  4  6  7  5  4  5    10  9  7 10  6  4  3  5  3  3
 38      1   2   5  20  70 102 174 209 266     3  21     4  4  4  3  5  5  6  6  5    15  5  5  5  6  5  5  3  6  5
 39      1   2   6  14  62 126 165 200 262     1  21     4  4  5  6  4  5  4  5  5     9  3  8  6  4  4  7  6  7  6
 40      1   2   5  20  65 103 161 225 260     1  22     2  4  3  5  6  7  4  7  6     9  8  9  8  5  6  1  2  4  4
 41      1   2   4  20  61 104 161 225 260     1  22     2  3  4  6  4  6  5  7  7     9  8  8  9  3  5  4  2  6  2
 42      1   2   4  20  64 116 164 220 242     1  22     2  4  3  6  7  6  5  6  5    10  8  8  7  7  2  3  4  4  3
 43      1   3  10  14  52 124 172 215 255     1  22     8  2  4  6  4  6  4  4  6    10  3  8  4  4  8  5  5  8  1
 44      1   2   5  20  66 103 160 225 267     2  22     2  4  4  4  7  7  4  6  6    14  8  5  5  4  3  3  4  4  6
 45      1   2   6  14  62 117 172 200 268     1  22     2  6  5  5  4  4  7  7  4     7  7  8  4  2  1  8  5 10  4
 46      1   2   6  22  61 117 167 220 232     1  22     2  6  4  6  4  4  5  6  7     6  7  6  2  7  4  8  4  7  5
 47      1   2   4  20  77 116 147 220 253     1  22     2  4  5  5  6  6  4  5  7     9 10  9  6  4  2  4  3  5  4
 48      1   2   4  20  74 117 156 225 243     1  22     2  4  6  5  6  5  5  5  6     9 12  8  5  5  2  6  5  2  2
 49      1   3  10  14  50 119 177 219 254     4  22     8  6  3  3  6  4  5  4  5     6  6  6  6  7  7  7  7  2  2
 50      1   2   6  14  65 119 177 217 250     1  22     4  5  5  6  6  3  6  6  3    10  1  5  9  3  3  7  6  4  8
 51      1   2   4  25  74 117 152 225 241     2  22     4  6  6  6  6  3  5  3  5     8  6 12  6  3  3  4  5  4  5
 52      1   2   4  20  57 123 174 218 252     1  22     4  3  5  4  4  6  5  6  7     9 11  6  5  7  7  2  3  4  2
 53      1   2   7  22  64 116 172 179 267     2  22     4  6  4  3  4  6  6  6  5     7  7  9  3  4  4  3  8  6  6
 54      1   2   6  22  64 130 137 211 269     1  22     4  6  4  6  6  4  5  6  3     6  7  3  6  4  6  3 10  6  5
 55      1   2   4  30  88 105 168 191 263     1  22     4  3  5  6  5  5  5  5  6     5  9 10  7  5  4  4  4  6  2
 56      1   2   6  14  62 129 150 218 265     1  22     4  5  4  6  4  4  5  6  6     7  5  8  5  2  3  6  5  9  6
 57      1   2   6  14  69 112 177 207 265     1  22     4  5  4  6  5  4  6  4  6     5  8  6  6  4  2  4  7  6  8
 58      1   2   4  20  57 124 172 207 260     2  22     4  4  6  4  4  5  6  5  6    12 10  5  3  6  6  4  3  4  3
 59      1   2   6  14  78 112 176 212 256     1  22     4  4  6  5  4  4  7  6  4     2  8  7  6  4  5  5  6  6  7
 60      1   2   6  17  61 113 176 201 270     1  22     4  4  6  6  4  4  6  6  4    11  5  5  4  4  3  7  5  6  5
 61      1   2   4  20  88 104 159 213 266     2  23     0  4  4  6  7  6  6  6  7     9  7  9  7  3  3  5  2  5  2
 62      1   2   4  20  64 114 161 225 243     1  23     2  3  5  6  7  6  4  7  6     9  9  6  8  5  4  4  2  4  1
 63      1   2   4  20  59 117 164 221 243     1  23     2  3  5  6  7  7  5  5  6     9  8  6  8  5  4  4  5  1  2
 64      1   2   4  20  71 104 161 225 253     1  23     2  3  5  5  6  6  6  7  6     8  8  8  8  3  5  4  1  5  2
 65      1   3  10  17  70 128 170 202 238     1  23     8  2  4  5  6  6  4  6  5    10  2  7  4  5  3  4  7  2  8
 66      1   3  10  17  74 131 142 206 261     2  23     8  2  5  4  4  6  6  6  5     4  6  8  8  6  2  4  6  2  6
 67      1   2   4  20  88 114 145 206 266     1  23     2  4  5  6  7  4  6  7  5     8  9  7  8  4  2  5  3  5  1
 68      1   2   6  22  64 116 162 223 232     1  23     2  7  7  5  4  5  6  4  6     6  5  7  4  6  2  6  6  5  5
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES BY FACTOR NUMBER

SCHED   *   *   *   *    FACTORS     *   *   *   * ORDER  HC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

  69    1   2   4  20   59 117 164 220 244     1   23     2  5  4  6  6  7  5  5  6      10  6  7  8  4  4  4  5  3  1
  70    1   2   6  22   61 117 162 223 232     1   23     2  6  5  5  5  4  6  6  7       7  6  5  2  8  3  6  5  6  4
  71    1   2   5  20   64 131 162 213 242     1   23     4  3  3  4  7  7  7  4  7      10  7  8  7  4  6  1  4  2  3
  72    1   3  10  15   83 122 142 205 264     1   23     8  4  6  6  6  4  5  3  4       7  6  8  3  7  2  6  5  6  2
  73    1   2   5  20   73 131 143 206 266     1   23     4  4  5  3  6  5  7  6  6       8  7  7 10  5  4  4  1  2  4
  74    1   2   6  14   62 124 172 200 262     1   23     4  6  5  7  4  6  5  6  3       8  4  6  5  2  3  5  6  7  6
  75    1   2   6  14   78 112 176 203 263     1   23     4  6  7  5  3  4  6  6  5       3  7  6  5  3  5  3  7  5  8
  76    1   2   6  14   59 109 169 220 262     1   23     4  6  3  6  4  5  5  7  6       7  6  5  4  4  3  6  5  4  8
  77    1   2   6  33   75 126 153 181 255     1   23     4  3  7  5  7  5  5  5  5       7  4  8  6  1  2  6  6  3  9
  78    1   2   6  17   65 118 177 200 262     1   23     4  6  6  5  5  6  6  3  5       6  4  7  4  5  3  3  9  5  6
  79    1   3  10  14   52 119 175 218 255     1   23     8  4  4  5  4  4  5  6  6       6  6  7  7  4  5  1  6  5  5
  80    1   2   6  14   76 125 160 200 263     1   23     4  5  6  7  4  4  5  4  7       6  7  6  6  3  2  7  5  7  3
  81    1   2   4  20   59 124 174 210 244     1   23     4  5  4  4  5  7  5  5  7       8  7  9  6  5  6  1  5  4  1
  82    1   2   6  17   75 129 155 218 251     1   23     4  7  4  6  5  4  6  5  5       6  5  7  4  3  1  7  6  7  6
  83    1   2   5  17   80 113 177 207 247     1   23     4  6  5  7  4  5  5  4  6       6  5  7  3  7  1  5  7  6  5
  84    1   2   5  23   58 102 169 224 260     1   23     4  5  4  5  7  5  4  6  6       8  7  8  9  4  3  2  4  4  3
  85    1   2   6  17   65 118 176 200 263     1   23     4  6  7  4  4  5  6  5  5       6  4  7  5  4  4  5  7  2  8
  86    1   2   6  17   90 113 169 207 247     1   23     4  4  5  7  4  6  5  6  5       8  5  7  2  4  4  6  5  4  7
  87    1   2   4  20   61 129 143 218 265     1   23     4  4  5  4  5  6  6  6  6       7 10  8  4  3  5  4  2  4  5
  88    1   2   4  25   77 117 152 221 241     1   23     4  5  6  7  5  4  5  5  5       6  7  8  9  4  1  4  3  4  6
  89    1   2   6  17   88 118 159 218 248     1   23     4  5  5  6  5  6  4  6  5       5  7  6  4  3  4  7  6  3  7
  90    1   2   5  26   80 116 149 223 231     1   23     4  6  5  5  5  5  5  5  6       8  6  4  5  5  3  6  5  7  2
  91    1   3  10  17   52 118 175 218 255     1   24     8  4  4  4  2  6  7  7  6       9  3  5  6  5  5  2  6  3  4
  92    1   2   6  22   65  95 159 218 265     2   24     2  6  4  7  4  7  5  7  6       6  6  3  6  3  3  6  6  3  6
  93    1   2   6  22   65 115 176 182 266     1   24     2  7  7  4  6  6  6  6  4       5  8  4  4  1  6  6  4  4  6
  94    1   3  10  17   66 127 169 189 257     1   24     8  4  5  2  6  5  7  5  6       8  4  5  4  7  0  9  1  5  5
  95    1   2   4  25   74 116 147 223 250     2   24     2  4  6  7  7  6  5  6  5      10  5 10  5  1  1  3  5  3  5
  96    1   2   4  20   88 116 143 208 266     1   24     2  4  6  6  5  5  6  7  7       9  7  7  7  3  4  5  3  2  1
  97    1   2   6  22   65 115 176 184 265     1   24     2  7  5  6  5  7  6  6  4       6  7  3  7  1  3  6  4  5  6
  98    1   2   4  20   57 117 165 225 252     2   24     2  4  6  5  5  6  7  6  7      10 10  0  4  4  4  3  2  3  2
  99    1   2   5  20   65 115 174 192 266     1   24     2  5  5  4  6  6  7  7  6      11  6  5  6  4  3  3  2  4  4
 100    1   2   4  20   64 116 143 225 265     2   24     2  5  6  5  7  5  7  6  5       9  9  6  4  3  3  2  2  6  4
 101    1   2   6  22   64 116 167 220 232     2   24     2  7  6  6  5  5  5  6  6       4  4  5  5  6  4  5  5  4  5
 102    1   2   9  31   78 131 151 202 227     1   24     6  7  3  5  5  3  7  7  5       5  3  5  9  1  9  6  5  3  2
 103    1   3  12  43   58 124 136 217 238     1   24     8  6  5  5  3  6  6  6  3       4  4  8  3 10  2  1  6  5  5
 104    1   3  12  36   84 119 137 189 253     1   24     8  5  6  3  6  6  3  5  6       4  2  2  7  5  7 10  2  6  3
 105    1   2   6  17   85 123 158 201 259     2   24     4  4  7  7  5  4  3  7  7       2  7  5  7  5  2  5  5  5  5
 106    1   2   4  20   74 116 174 208 242     1   24     4  3  4  5  7  7  6  7  5       9  8  7  4  5  3  4  4  1  3
 107    1   2   5  14   62 124 172 217 250     1   24     4  5  5  7  4  7  6  7  3       7  2  7  5  2  2  5  6  7  5
 108    1   2   6  17   75 113 177 201 255     1   24     4  5  7  7  4  3  7  5  6       6  6  6  6  2  2  7  5  6  2
 109    1   3  12  36   79  92 172 204 237     1   24     8  4  6  4  6  6  3  5  6       3  4  3  6  1  7  5  4 10  5
 110    1   3  10  17   77 131 142 202 261     1   24     8  4  5  3  6  6  5  5  6       5  5  7  8  5  3  1  6  3  5
 111    1   3  12  29   83  96 167 215 238     1   24     8  6  6  5  6  5  5  3  4       4  6  6  6  4  5  1  7  5  4
 112    1   3  11  15   64 131 163 216 235     1   24     8  5  6  3  6  6  5  5  4       6  7  6  2  6  4  6  3  4  4
 113    1   2   4  25   77 117 172 197 243     1   24     4  5  6  5  7  5  7  3  6       6  8  7  8  1  5  4  5  3  1
 114    1   2   6  24   76 115 172 207 230     2   24     4  7  3  7  6  5  5  5  6       6  6  2  6  6  2  4  4  6  6
 115    1   2   4  28   78 110 176 191 259     1   24     4  5  3  5  6  7  6  7  5       5  9  9  5  2  3  3  4  4  4
 116    1   2   6  17   71 118 160 225 248     1   24     4  3  7  7  5  5  5  6  6       6  4  7  5  3  3  4  5  6  5
 117    1   2   4  23   78 110 172 151 262     1   24     4  5  3  6  7  6  6  6  5       8  6  9  7  0  3  3  3  4  5
 118    1   2   8  23   58 114 169 224 230     1   24     4  6  6  5  6  3  5  6  7       9  6  4  5  5  3  3  5  2  6
 119    1   2   6  17   78 117 177 202 248     1   24     4  6  5  6  6  3  7  5  6       6  5  3  6  2  4  7  6  5  4
 120    1   2   7  15   67 111 165 225 257     1   24     6  5  5  6  3  7  6  5  5       9  3  1  8  2  5  3  4  5  8
 121    1   3  10  14   52 129 169 203 261     1   24     8  4  6  4  4  6  5  4  7       8  3  6  7  5  3  3  4  4  5
 122    1   2   4  20   88 124 145 204 259     1   24     4  4  5  4  7  6  7  4  7       7  9  8  4  4  4  5  3  3  1
 123    1   2   4  20   77 116 172 194 253     1   24     4  4  5  4  7  6  7  4  7       7  9  8  6  4  2  6  2  2  2
 124    1   3  10  17   70 118 175 209 238     1   24     8  4  5  5  5  4  7  6  4      10  3  4  5  2  6  6  3  3  6
 125    1   2   6  22   73 118 137 224 256     2   24     4  5  4  5  7  7  7  5  4       6  6  3  6  6  4  3  5  5  4
 126    1   3  12  42   60  92 175 217 234     1   24     8  6  6  5  4  4  6  4  5       3  5  7  5  6  5  1  8  2  6
 127    1   3  12  42   60 126 134 216 235     1   24     8  6  6  4  5  6  4  4  5       1  6  6  7  5  5  4  6  3  5
 128    1   3  10  17   70 125 175 202 238     1   24     8  4  4  5  6  6  5  4  6      11  2  4  6  3  3  4  6  5  4
 129    1   2   6  17   80 115 172 207 247     1   24     4  7  4  7  5  5  6  4  6       5  4  4  5  7  2  4  7  5  5
 130    1   2   4  20   74 129 156 218 243     1   24     4  4  6  4  7  6  6  6  5       7 10  8  5  4  3  4  2  2  3
 131    1   2   6  26   73 109 168 221 227     1   24     4  5  4  7  6  6  6  4  6       4  7  3  7  2  2  5  4  8  6
 132    1   2   4  30   78 105 177 191 262     6   24     4  4  4  6  6  6  6  6  6       8  6  8  8  3  3  3  3  3  3
 133    1   3  12  36   79  92 172 188 253     1   24     8  5  6  4  5  5  4  6  5       1  5  2  6  3  6  6  4 10  5
 134    1   3  10  17   77 128 142 202 264     1   24     8  4  5  4  5  6  5  5  6       5  5  7  8  5  2  1  6  4  5
 135    1   2   5  23   55 122 172 209 260     2   24     4  6  4  5  5  7  5  7  5       8 12  5  3  5  5  2  3  2  3
 136    1   2   5  23   58 102 176 218 267     2   24     4  6  4  5  7  5  5  7  5       8  6  8  8  2  4  3  3  4  2
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES BY FACTOR NUMBER

SCHED    *   *   *   *    FACTORS    *   *   *   *  ORDER  HC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

 137     1   2   4  25  77 117 150 220 244     1   24    4  5  5  5  7  6  6  4  6     7  8  5  8  0  5  6  2  3  4
 138     1   2   6  17  75 129 155 201 263     1   24    4  7  6  5  4  6  5  6  5     7  5  5  5  4  1  5  4  8  4
 139     1   2   6  17  68 118 160 225 251     1   24    4  5  6  7  6  4  5  6  5     6  5  5  7  2  2  6  6  4  5
 140     1   2   4  25  74 116 154 225 241     1   24    4  5  5  6  6  6  4  5  7     8  6  9  4  4  3  2  5  3  4
 141     1   2   6  17  80 116 172 206 247     1   24    4  6  5  7  5  4  6  5  6     5  6  2  7  4  5  6  7  3  3
 142     1   2   6  17  78 112 177 202 262     1   24    4  6  5  7  6  4  6  5  5     4  5  5  6  2  3  4  7  5  7
 143     1   2   6  14  78 109 176 208 259     1   24    4  6  6  7  5  5  4  5  6     4  7  4  6  3  3  6  4  3  8
 144     1   2   6  17  66 117 152 206 266     1   24    4  6  6  6  6  4  5  5  6     6  5  5  4  6  1  5  6  7  3
 145     1   2   6  17  78 112 176 202 263     1   24    4  6  6  6  4  5  6  6  5     4  5  5  7  3  2  2  7  6  7
 146     1   2   6  24  76 113 177 182 255     1   24    4  5  6  6  6  4  6  6  5     9  4  4  5  6  2  4  5  5  4
 147     1   2   7  22  45 116 169 220 254     2   24    4  5  5  5  7  6  5  6  5     5  5  6  3  3  6  4  6  6  4
 148     1   2   5  23  58 129 144 218 267     2   24    4  6  6  6  5  5  5  5  6     6  8 10  4  4  3  3  3  3  4
 149     1   2   6  17  90 111 155 217 270     1   24    4  5  6  5  5  6  6  6  5     9  5  3  4  4  3  3  7  5  5
 150     1   3  11  15  53 118 176 209 263     4   24    8  5  5  5  5  5  5  5  5     8  8  4  4  4  4  4  4  4  4
 151     1   2   6  22  65  94 168 220 263     1   25    2  6  7  5  6  4  7  7  6     6  4  2  5  5  4  3  2  7  6
 152     1   2   6  17  65 115 176 200 266     1   25    2  7  7  6  4  6  7  5  6     5  5  6  3  2  3  5  6  3  6
 153     1   3  10  17  74 128 142 206 264     2   25    8  2  5  5  5  6  6  6  7     4  4  7  7  6  1  4  4  1  6
 154     1   2   6  22  65 130 135 217 266     1   25    2  7  6  6  6  6  6  6  5     5  5  5  3  3  5  3  2  9  4
 155     1   3  12  42  69 126 134 206 235     1   25    8  5  6  6  7  7  4  3  4     1  8  3  8  6  4  6  3  2  3
 156     1   3  10  14  52 129 166 208 259     1   25    8  3  6  6  4  6  7  6  4     6  4  5  4  5  6  7  2  4  1
 157     1   3  10  15  83  97 169 215 255     1   25    8  3  7  5  7  6  5  4  5     7  5  9  3  2  2  1  7  4  4
 158     1   2   6  17  71 112 167 221 255     1   25    4  3  5  7  7  6  6  5  7     5  5  6  5  0  3  3  4  7  6
 159     1   2   7  17  88 112 161 204 263     1   25    4  3  6  7  7  5  5  6  7     4  4  5  9  2  6  1  3  7  3
 160     1   3  10  15  67 131 169 189 253     1   25    8  3  6  6  6  4  5  6  6     4  7  6  5  4  2  5  2  2  7
 161     1   2   7  21  46 117 163 222 256     1   25    4  3  6  5  7  6  7  6  6     6  8  4  3  4  4  3  4  6  2
 162     1   2   5  25  66 122 176 155 251     1   25    6  6  3  5  6  6  5  6  7    10  4  7  5  4  3  2  3  2  4
 163     1   2   5  14  60 126 169 217 251     1   25    6  5  5  7  6  6  3  6  6     7  3  6  3  3  2  6  7  3  4
 164     1   3  10  19  66 122 178 205 231     1   25    8  6  4  7  5  4  8  4  4     6  4  7  1  1  8  9  1  2  5
 165     1   3  13  26  74 130 138 217 235     2   25    8  5  7  7  4  5  6  4  4     2  5  5  3  5  2  7  7  5  3
 166     1   3  10  17  74  99 175 218 255     1   25    8  4  4  6  5  4  6  7  6     7  2  5  5  6  1  5  6  2  5
 167     1   3  10  14  52 126 166 218 255     1   25    8  4  6  7  4  6  5  6  4     7  7  5  4  4  2  3  4  5  3
 168     1   2   6  22  64 131 137 211 268     1   25    4  7  6  4  6  6  7  6  4     5  5  3  8  4  3  3  5  3  5
 169     1   3  10  15  63  99 166 218 255     1   25    8  5  7  5  7  4  5  4  5     7  5  5  4  6  1  4  5  6  1
 170     1   2   6  14  62 106 176 218 265     1   25    4  5  5  7  4  5  7  7  6     6  5  6  4  0  4  6  2  6  5
 171     1   2   6  22  64 131 137 216 265     1   25    4  7  4  5  7  6  7  5  5     6  4  5  6  2  3  5  3  4  6
 172     1   3  10  14  50 119 177 209 261     1   25    8  6  5  4  6  4  5  6  6     7  7  3  5  3  4  4  7  2  2
 173     1   3  10  14  52 124 175 212 255     1   25    8  4  4  6  5  6  5  6  6     6  4  8  4  4  5  2  4  4  3
 174     1   2   6  17  90 123 155 218 247     1   25    4  5  5  7  4  6  6  7  6     7  4  4  5  1  6  5  5  4  3
 175     1   2   6  37  61  97 162 224 249     1   25    4  5  4  6  6  5  7  7  6     5  5  7  3  2  3  4  3  5  7
 176     1   2   6  14  78 109 176 218 252     1   25    4  7  5  7  4  5  6  6  6     3  5  4  6  2  4  6  3  4  7
 177     1   2   7  22  45 116 172 208 267     1   25    4  5  4  5  7  7  6  6  6     5  7  4  3  4  2  4  5  5  3
 178     1   2   6  17  86 116 154 206 266     1   25    4  5  7  7  6  4  6  5  6     5  5  4  4  6  2  3  6  3  6
 179     1   2   6  17  75 117 172 206 251     1   25    4  6  4  6  6  5  6  6  7     5  6  3  5  3  1  8  5  5  3
 180     1   2   6  17  65 118 152 225 262     1   25    4  6  6  6  6  4  6  5  7     6  3  5  5  4  2  3  7  5  4
 181     1   2   4  25  77 116 172 194 247     1   25    4  5  5  7  7  5  7  5  5     5  9  7  7  2  2  5  4  2  1
 182     1   3  13  26  74  97 177 217 235     2   25    8  5  6  6  4  6  5  5  5     2  6  3  4  6  2  7  7  3  4
 183     1   2   6  14  76 130 160 200 259     1   25    4  6  5  7  5  6  5  5  7     4  6  6  4  2  3  5  6  6  2
 184     1   2   6  17  68 116 169 202 248     1   25    4  5  5  7  5  6  6  6  6     6  4  4  6  0  5  5  6  3  5
 185     1   2   4  25  71 104 176 199 266     1   25    4  5  5  6  6  5  7  6  6     8  5  6  7  4  1  3  2  5  3
 186     1   2   6  22  64  95 168 220 263     1   25    4  6  7  5  5  5  6  6  6     7  4  2  7  3  4  3  4  5  5
 187     1   2   6  22  64  96 177 211 269     1   25    4  6  6  5  6  5  7  6  5     5  7  3  5  2  4  4  6  5  3
 188     1   2   6  24  60 130 168 211 232     1   25    4  6  6  6  7  6  5  5  5     4  6  4  3  4  5  2  8  4  4
 189     1   2   8  23  85 109 164 182 259     1   25    4  6  6  6  5  6  5  6  6     3  5  8  7  4  2  4  3  3  5
 190     1   2   7  24  60 130 165 180 259     1   25    6  5  6  5  5  6  7  5  5     5  3  6  5  6  5  1  6  2  5
 191     1   2   6  17  65 116 162 223 247     1   26    2  7  7  7  4  6  7  7  5     4  5  3  6  2  2  4  6  5  3
 192     1   2   6  24  60 116 162 223 232     2   26    2  7  7  5  7  5  7  5  7     5  5  4  4  3  3  5  5  3  3
 193     1   3  10  17  74 128 141 218 255     2   26    8  2  5  6  6  5  6  8  6     6  4  4  4  6  0  6  4  0  6
 194     1   2   6  22  80  94 159 223 259     2   26    2  7  5  6  7  5  6  7  7     6  6  1  1  5  3  5  3  5  5
 195     1   3  10  17  70 125 172 189 253     2   26    8  3  4  5  6  8  5  6  7     6  3  5  6  5  3  5  1  5  1
 196     1   3  13  38  68  97 178 206 233     1   26    8  5  7  4  3  7  7  5  6     2  6  2  7  4  0  3  8  5  3
 197     1   3  11  15  64 131 167 215 233     1   26    8  5  6  4  7  6  7  3  6     3  7  8  0  4  5  2  6  3  2
 198     1   3  10  14  60 125 175 185 263     1   26    8  6  4  6  6  7  3  6  6     7  4  5  4  4  2  4  4  1  5
 199     1   3  11  25  81  99 176 179 259     2   26    8  7  6  6  6  6  3  6  4     5  5  2  5  2  5  3  5  3  5
 200     1   3  12  25  83 124 137 189 262     1   26    8  6  8  6  5  5  3  5  6     3  3  6  2  5  6  7  2  2  4
 201     1   3  10  31  91 119 142 206 229     2   26    8  3  6  6  7  5  5  5  7     6  3  4  4  2  5  6  3  5  7
 202     1   2   7  41  47 108 173 212 255     2   26    6  3  5  7  5  7  7  6  6     4  4  4  4  5  5  1  6  1  6
 203     1   2   7  24  47 130 165 212 255     2   26    6  3  6  7  6  7  7  5  5     6  6  3  3  3  3  2  6  2  6
 204     1   2   7  24  47 129 163 214 256     2   26    6  3  7  5  7  6  6  6  6     6  6  2  2  4  4  5  3  5  3
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES BY FACTOR NUMBER

SCHED      *   *   *   *    FACTORS    *   *   *   * ORDER  HC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

 205       1   3  10  18  67 128 169 183 255     1  26     8  4  6  8  6  5  5  6  4     7  3  6  4  2  2  2  5  1  8
 206       1   3  13  25  74  96 176 189 263     1  26     8  6  7  5  4  7  4  6  5     7  3  1  6  4  4  4  4  2  5
 207       1   2   6  35  64  96 162 224 249     1  26     4  5  7  4  7  7  7  6  5     4  5  5  5  1  2  3  4  3  8
 208       1   3  10  15  72 131 163 189 253     1  26     8  4  6  7  6  5  6  6  4     5  6  5  5  4  3  4  2  0  6
 209       1   3  10  15  67 127 169 189 257     1  26     8  4  7  6  4  5  6  6  6     6  4  4  6  4  1  5  2  2  6
 210       1   3  10  14  60 124 172 186 264     1  26     8  4  5  7  6  6  4  6  6     7  4  3  5  1  5  4  4  4  3
 211       1   3  10  15  83 122 142 215 257     1  26     8  4  7  6  6  6  5  4  6     4  6  6  5  4  3  3  5  2  2
 212       1   2   6  17  65 108 162 225 262     1  26     4  7  6  7  4  5  6  7  6     5  6  2  5  2  3  4  5  5  3
 213       1   2   6  17  64 113 168 225 251     1  26     4  5  6  7  7  6  4  7  6     7  5  3  4  2  2  5  4  4  4
 214       1   3  11  25  70 126 170 187 231     2  26     8  6  6  4  4  6  6  6  6     6  2  5  5  4  4  3  3  4  4
 215       1   2   6  17  90 106 160 216 266     1  26     4  4  6  6  6  6  6  7  7     6  5  5  2  1  5  3  5  4  4
 216       1   2   6  14  60 112 177 217 265     1  26     4  6  6  6  6  4  7  7  6     5  6  4  5  1  3  4  3  4  5
 217       1   2   6  22  65 130 162 184 262     1  26     4  7  4  6  6  7  6  6  6     5  4  4  4  4  2  2  5  7  3
 218       1   3  10  19  70 129 163 183 254     1  26     8  6  5  4  6  5  8  5  5     7  3  4  3  3  2  8  1  7  2
 219       1   3  12  36  84 119 136 204 238     1  26     8  5  5  5  7  6  7  4  5     7  1  4  4  3  3  4  3  7  4
 220       1   3  13  38  69  96 178 206 233     1  26     8  6  6  4  5  5  6  5  7     4  3  3  6  5  0  6  5  5  3
 221       1   2   6  14  86 117 163 200 256     1  26     4  6  5  6  7  5  7  7  5     3  6  4  4  3  1  6  5  6  2
 222       1   2   6  22  71  95 176 208 266     1  26     4  5  7  5  7  7  6  6  5     5  7  3  3  2  5  5  1  4  5
 223       1   2   6  24  60  95 168 220 263     1  26     4  6  7  5  6  5  7  5  7     5  6  3  5  1  4  4  2  4  6
 224       1   2   6  26  46 117 177 198 265     1  26     4  7  5  5  6  6  7  7  5     5  6  1  4  3  2  5  4  4  6
 225       1   3  11  18  58 128 169 183 262     1  26     8  6  6  6  4  5  5  6  6     3  6  4  5  3  4  4  3  2  6
 226       1   2   6  14  65 119 176 200 263     1  26     4  6  6  7  5  5  6  6  7     4  3  6  5  2  1  5  5  2  7
 227       1   2   6  30  67 122 152 224 231     1  26     4  5  7  5  6  6  7  6  6     4  5  4  3  1  4  4  4  4  7
 228       1   2   6  37  60 108 162 224 227     1  26     4  7  6  6  7  5  6  5  6     6  3  4  4  1  4  5  4  4  5
 229       1   2   5  23  58 125 144 225 260     1  26     4  5  5  6  7  6  7  6  6     5 10  5  4  3  4  2  2  3  2
 230       1   2   6  24  59 131 138 211 268     1  26     4  6  6  7  6  7  5  6  5     3  4  2  5  6  4  5  2  4  5
 231       1   2   6  22  85  96 162 218 257     1  26     4  6  6  5  6  7  6  5  7     5  5  6  3  2  3  3  6  3  4
 232       1   2   5  17  75 115 172 201 255     1  26     4  6  6  7  5  6  6  6  6     5  5  5  5  1  2  5  5  5  2
 233       1   2   6  22  64 130 162 217 232     1  26     4  7  6  5  6  6  6  6  6     5  3  5  3  4  3  3  7  5  2
 234       1   2   5  23  55 122 174 209 259     4  26     4  6  6  6  6  6  6  6  6    10  8  3  3  4  4  2  2  2  2
 235.      1   3  12  35  84 120 136 203 238     1  26     8  6  5  5  5  5  7  5  6     5  3  6  2  3  5  5  3  6  2
 236       1   2   7  34  60 124 137 222 245     1  26     6  5  5  7  5  7  5  5  7     6  4  2  6  1  5  2  4  5  5
 237       1   2   8  25  85 125 152 179 259     1  26     6  5  6  7  5  5  6  5  7     1  6  5  7  4  4  3  4  1  5
 238       1   2   7  17  77 127 159 201 259     1  26     6  5  7  6  6  5  5  5  7     5  7  3  3  1  5  5  3  4  4
 239       1   2   6  17  61 129 162 218 248     1  26     6  5  5  6  5  6  7  5  7     4  6  5  2  2  3  5  3  6  4
 240       1   2   7  24  77  97 172 214 257     1  26     6  5  5  6  7  5  6  5  7     7  3  3  4  3  6  3  4  4  3
 241       1   2   7  15  80 107 171 204 263     1  26     6  5  5  6  6  6  5  6  7     5  5  4  4  3  6  0  6  4  3
 242       1   2   6  14  69 123 177 201 252     1  26     6  5  6  6  5  6  6  5  7     2  5  5  6  4  2  2  5  4  5
 243       1   3  12  44  67  92 170 204 238     2  27     8  4  7  7  8  6  6  6  2     6  0  6  0  2  4  2  6  4  6
 244       1   2   6  14  73 109 172 207 268     1  27     4  7  3  7  7  6  7  7  6     4  5  3  4  2  0  4  3  6  5
 245       1   2   6  28  73 122 137 224 245     1  27     4  7  3  6  7  7  7  7  6     3  4  5  4  3  2  1  5  6  3
 246       1   3  10  17  81 131 140 206 259     1  27     8  3  5  6  5  6  8  7  6     3  5  4  5  5  1  4  2  3  4
 247       1   3  12  25  63  96 170 216 238     1  27     8  6  7  6  5  7  3  7  5     6  2  7  2  3  3  0  6  2  5
 248       1   3  10  17  77  99 172 219 253     1  27     8  5  3  5  7  7  6  6  7     6  2  4  4  4  3  6  1  3  3
 249       1   3  10  15  83 121 142 206 264     1  27     8  4  7  5  6  7  7  4  6     6  3  5  4  6  0  5  2  3  2
 250       1   3  10  15  83 121 141 218 255     1  27     8  4  7  5  6  7  4  6  7     6  3  5  4  5  1  5  2  4  1
 251       1   3  11  15  64 131 167 212 235     1  27     8  5  7  4  6  7  7  6  4     4  6  5  1  5  3  4  3  1  4
 252       1   3  10  17  89 125 140 202 263     1  27     8  6  4  4  7  7  7  5  6     8  2  3  6  4  2  2  3  2  4
 253       1   3  12  44  63 118 134 215 238     1  27     8  4  7  5  6  6  7  7  4     4  2  6  2  3  4  6  3  3  3
 254       1   3  10  32  81  95 177 185 255     2  27     8  6  4  6  4  8  6  6  6     3  2  7  2  5  1  6  1  3  6
 255       1   3  12  39  63 127 134 185 259     1  27     8  4  8  7  5  6  5  5  6     1  4  1  7  5  3  8  1  6  0
 256       1   3  10  32  89 118 142 180 256     1  27     8  5  5  6  7  8  6  5  4     7  1  2  7  6  2  2  3  2  4
 257       1   3  11  15  84 125 141 212 253     1  27     8  6  5  4  5  6  5  8  7     2  5  6  3  5  1  4  5  3  2
 258       1   3  10  18  72 129 163 183 254     1  27     8  7  5  7  4  5  7  6  5     5  3  6  3  2  2  5  2  3  5
 259       1   3  12  36  89  92 163 204 237     2  27     8  5  7  4  5  7  7  5  6     2  4  3  3  2  4  3  3  6  6
 260       1   3  11  25  91 124 133 187 259     1  27     8  6  6  5  5  7  4  6  7     6  1  2  3  5  3  2  4  7  3
 261       1   3  12  42  69  92 175 207 234     1  27     8  7  7  6  6  4  6  5  5     2  4  5  3  6  3  3  3  2  5
 262       1   3  10  32  81  99 172 179 255     1  27     8  5  4  6  6  7  6  7  5     4  2  5  4  4  2  4  4  4  3
 263       1   2   6  17  65 129 162 197 263     1  27     4  7  6  5  5  7  7  6  7     5  2  3  6  3  1  4  4  5  3
 264       1   2   6  26  46 117 163 223 251     1  27     4  7  5  5  7  6  7  6  7     7  4  2  3  4  1  3  4  4  4
 265       1   2   6  14  60 116 169 220 248     1  27     4  5  6  7  7  7  5  6  7     5  5  3  4  1  2  4  4  3  5
 266       1   3  10  14  52 129 163 208 261     1  27     8  4  6  6  5  6  6  6  7     5  3  6  4  6  3  4  1  3  1
 267       1   3  10  15  87  99 163 219 253     1  27     8  6  4  6  6  7  6  5  6     4  5  7  3  2  1  4  4  3  3
 268       1   3  10  17  91 122 140 209 259     1  27     8  4  6  6  7  5  6  6  6     7  3  3  4  4  3  4  1  5  2
 269       1   3  12  36  89  92 163 188 253     2  27     8  6  7  4  6  6  6  6  5     2  3  2  5  2  3  2  5  6  6
 270       1   3  10  17  89 118 140 218 257     1  27     8  6  6  4  6  7  6  5  6     5  3  7  3  2  3  3  3  2  5
 271       1   2   6  17  90 122 160 197 259     1  27     4  5  6  7  6  7  6  6  7     7  2  4  3  1  3  4  6  3  3
 272       1   2   6  22  73 118 137 223 257     1  27     4  6  5  7  7  7  6  6  6     6  4  2  3  5  2  2  5  4  3
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES BY FACTOR NUMBER

SCHED    *  *  *  *   FACTORS .  *   *   *   * ORDER  FC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES
 273     1  2  6  22  65 118 177 182 262          1   27    4  6  6  6  7  7  7  6  5             5  5  4  2  4  2  2  4  4  4
 274     1  3 10  31  79 121 142 202 227          1   27    8  5  5  5  8  5  6  7  5             4  2  5  5  3  4  2  5  2  4
 275     1  3 10  34  79 128 142 202 227          1   27    8  5  5  5  6  5  7  6  7             5  3  5  5  3  1  1  4  5  4
 276     1  2  8  16  61 126 164 201 262          1   27    6  5  5  7  7  5  7  5  7             8  1  4  6  3  3  4  1  5  1
 277     1  2  7  24  71 129 163 179 254          1   27    6  5  7  5  5  5  7  7  7             6  4  2  3  3  5  5  1  4  3
 278     1  2  6  17  73 118 177 207 248          1   27    6  5  5  7  7  5  7  5  7             4  5  3  3  3  2  4  4  5  3
 279     1  3 10  19  66 131 166 206 231          1   27    8  5  5  7  6  6  6  6  5             4  3  4  3  4  3  6  4  0  5
 280     1  3 12  43  63 119 136 217 238          1   27    8  5  3  6  6  6  7  6  5             4  1  4  4  6  1  2  4  2  6
 281     1  3 11  25  91  95 169 185 262          1   27    8  6  5  6  5  7  5  6  6             4  4  4  4  2  3  4  2  5  4
 282     1  2  7  17  78 131 148 218 253          1   27    6  5  5  7  6  6  5  7  7             4  3  3  6  0  4  5  3  4  4
 283     1  2  8  15  80 127 159 212 246          1   27    6  5  6  7  7  5  7  5  6             3  6  3  5  3  1  2  8  2  3
 284     1  2  6  17  59 118 177 201 262          1   27    6  6  5  7  5  6  7  7  5             3  5  4  2  4  1  4  6  2  5
 285     1  3 10  19  77 129 138 202 264          2   27    8  6  5  6  6  6  6  6  5             2  5  6  6  2  3  2  5  3  2
 286     1  2  6  25  46 124 176 212 251          1   27    6  6  6  6  5  6  7  5  7             5  2  3  5  2  4  3  4  2  6
 287     1  2  6  26  83 124 138 217 251          1   27    5  5  6  6  6  6  6  7  6             4  4  3  3  1  4  5  5  2  5
 288     1  2  6  25  46 119 176 201 263          1   27    6  6  6  7  6  6  5  6  6             8  2  3  3  3  2  3  4  3  5
 289     1  2  6  14  73 123 177 198 252          1   27    6  6  6  7  5  6  6  6  6             4  3  3  7  3  2  3  3  5  3
 290     1  2  5  26  73 121 177 196 240         54   27    6  6  6  6  6  6  6  6  6             3  9  3  3  3  3  3  3  3  3
 291     1  2  9  21  76 130 134 211 253          9   27    6  6  6  6  6  6  6  6  6             3  9  3  3  3  3  3  3  3  3
 292     1  3 10  14  79 131 140 208 259          2   28    8  3  7  7  7  6  7  5  6             3  4  3  4  4  4  4  4  1  1
 293     1  3 10  17  91 125 140 206 259          1   28    8  3  6  6  6  6  8  6  7             5  4  4  4  3  2  4  0  4  2
 294     1  3 12  42  60  93 175 216 234          1   28    8  8  6  4  4  7  7  5  7             3  1  6  3  4  3  2  5  1  4
 295     1  3 11  15  63 127 163 167 264          1   28    8  6  6  4  4  7  6  8  7             2  5  4  3  5  2  4  2  2  3
 296     1  3 13  38  59  97 178 216 233          1   28    8  7  6  4  4  7  7  6  7             1  3  3  4  4  1  4  5  2  5
 297     1  3 10  18  67 128 170 206 231          2   28    8  4  5  8  8  5  5  7  6             3  3  4  3  5  1  5  4  1  3
 298     1  3 10  19  89 118 133 205 263          1   28    8  6  5  8  8  5  6  4  6             6  3  4  2  3  3  4  1  4  2
 299     1  3 11  31  64 119 178 187 226          2   28    8  8  6  5  6  8  4  6  5             2  2  3  6  3  6  2  3  2  3
 300     1  3 10  17  64 122 172 186 264          1   28    8  4  5  6  8  7  5  7  6             6  3  4  4  2  2  2  3  4  2
 301     1  3 10  18  72 121 172 183 255          1   28    8  5  5  7  7  7  4  6  7             5  1  3  6  0  1  3  4  0  3
 302     1  3 10  26  91 121 133 218 234          1   28    8  6  7  5  4  7  7  7  5             4  1  5  5  3  3  3  1  5  6
 303     1  3 10  26  81 130 133 212 236          1   28    8  7  6  7  4  7  5  7  5             3  3  5  5  1  4  4  4  2  4
 304     1  3 12  44  72  93 166 202 238          1   28    8  7  4  5  7  7  6  7  5             1  3  5  2  1  4  3  2  5  4
 305     1  2  6  24  71  95 163 223 255          2   28    4  5  7  5  7  7  7  7  7             5  5  4  1  1  2  2  5  4  2
 306     1  3 10  27  67 124 175 180 238          1   28    8  6  5  8  6  4  7  6  6             3  3  4  5  4  2  1  3  3  3
 307     1  3 11  15  64 121 142 202 264          1   28    8  6  6  4  5  6  7  7  7             5  4  6  0  2  6  5  2  2  2
 308     1  3 11  29  91  97 167 166 259          1   28    8  5  7  6  6  7  7  4  6             5  1  6  1  2  4  2  2  5  3
 309     1  3 10  17  64 122 178 186 259          1   28    8  4  6  6  7  7  6  7  5             7  2  3  4  2  4  3  2  2  2
 310     1  3 10  19  91 121 133 206 259          1   28    8  4  7  6  6  7  7  5  6             5  2  4  4  [illegible]  3  4  2  3  2
 311     1  3 12  36  84  96 163 189 253          1   28    8  6  7  4  6  7  6  6  6             4  1  1  5  4  3  [illegible]  3  [illegible]  3
 312     1  3 10  14  60 124 178 166 255          1   28    8  4  6  7  7  6  6  6  6             4  5  4  3  1  3  [illegible]  2  4  [illegible]
 313     1  3 13  25  74  96 177 189 262          1   28    8  6  6  7  6  6  4  6  7             6  2  2  4  3  2  5  5  4  2
 314     1  2  5  27  86 117 163 200 227          1   28    4  7  6  7  7  6  7  6  6             2  6  4  3  1  1  3  5  2  3
 315     1  2  8  23  80  93 157 223 259          2   28    4  7  6  6  7  6  6  7  7             6  6  2  2  4  2  4  2  4  2
 316     1  2  6  24  86 115 163 182 255          2   28    4  6  7  6  7  7  6  7  6             5  5  2  2  2  2  4  3  2  2
 317     1  3 10  31  79 130 142 202 229          1   28    8  5  6  7  7  6  5  7  5             5  0  3  5  1  5  4  4  4  2
 318     1  3 10  15  87 121 142 202 264          1   28    8  5  5  6  6  7  7  5  7             6  2  5  3  3  0  3  4  2  4
 319     1  3 12  36  84 119 136 190 253          1   28    8  7  6  5  5  7  7  5  6             3  1  1  5  5  3  4  1  3  3
 320     1  3 10  18  81 128 132 212 255          1   28    8  5  6  7  6  5  7  7  5             4  4  5  1  4  1  4  2  4  2
 321     1  3 10  15  79 131 139 218 253          1   28    8  5  6  6  7  5  7  7  5             4  6  3  3  4  2  4  1  2  5
 322     1  3 12  29  83  96 166 216 238          1   28    8  7  6  5  7  6  5  7  5             2  4  5  3  2  4  4  4  2  3
 323     1  3 10  34  81 126 138 206 236          1   28    8  6  5  7  6  6  5  7  6             3  2  5  3  3  5  1  2  4  3
 324     1  3 11  25  70 125 172 183 237          2   28    8  6  6  6  6  5  5  7  7             4  0  5  5  3  3  6  2  0  4
 325     1  3 12  25  83  96 167 219 238          1   28    8  7  6  6  6  7  5  6  5             4  3  [illegible]  3  5  2  2  4  4  3
 326     1  3 11  15  53 118 177 205 262          1   28    8  5  6  6  5  6  7  6  7             5  7  3  3  2  3  0  3  3  1
 327     1  3 12  44  64 122 134 215 233          1   28    8  6  7  7  6  6  5  6  5             3  1  4  3  3  6  2  2  3  3
 328     1  3 10  14  50 121 178 218 252          8   28    8  6  6  6  6  6  6  6  6             4  4  4  4  4  4  4  4  0  0
 329     1  3 11  15  53 121 177 202 262          4   28    8  6  6  6  6  6  6  6  6             4  8  2  2  3  3  4  3  2  2
 330     1  3 11  18  48 128 169 207 262          8   28    8  6  6  6  6  6  6  6  6             4  4  3  3  3  3  3  3  3  3
 331     1  3 10  18  64 122 172 215 229          2   29    8  4  6  8  8  8  4  6  6             3  0  3  5  1  5  3  1  5  0
 332     1  3 12  44  58 120 132 219 238          2   29    8  7  4  7  6  7  7  8  4             2  1  6  1  4  2  5  6  4  1
 333     1  3 10  15  54 118 177 215 255          1   29    8  4  6  5  6  8  8  6  7             6  3  5  2  2  4  1  3  1  1
 334     1  3 10  15  72 131 166 202 238          1   29    8  4  6  7  8  5  7  7  6             5  2  4  2  4  0  1  5  7  2
 335     1  3 11  31  76 100 177 202 228          1   29    8  8  7  6  7  7  5  6  4             3  3  2  2  3  5  2  2  0  3
 336     1  3 10  15  81 129 139 205 263          1   29    8  5  4  7  7  7  7  6  7             4  3  5  3  3  3  5  0  1  3
 337     1  3 10  31  91 100 163 206 229          1   29    8  4  7  7  7  6  7  5  7             4  1  3  4  2  4  3  2  3  1
 338     1  3 10  26  81 130 140 212 229          1   29    8  6  6  8  6  6  7  7  4             1  3  5  4  3  3  4  2  2  2
 339     1  3 10  18  49 122 172 219 253          1   29    8  6  4  6  7  8  6  6  7             3  2  3  4  2  4  3  3  2  1
 340     1  3 12  39  72  93 172 189 253          1   29    8  6  6  7  6  7  4  7  7             1  4  3  3  2  3  4  0  3  5
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES BY FACTOR NUMBER

SCHED      *   *   *   *    FACTORS    *   *   *   * ORDER  HC    # HAMILT CIRCUITS ACROSS THE FCTRS   # 4-CIRCUITS ACROSS THE VERTICES

 341       1 . 3  12  42  59  92 176 215 224      1   29    8  6  7  6  7  6  7  7  4    3  2  4  3  4  2  2  2  4  2
 342       1   3  11  31  76 130 142 202 228      1   29    8  7  7  5  7  7  5  7  5    4  2  4  1  2  4  4  3  0  4
 343       1   3  12  32  64  96 170 202 238      1   29    8  6  6  6  8  7  5  7  5    3  2  6  1  1  3  1  5  1  5
 344       1   3  10  32  91 122 142 180 252      1   29    8  5  6  6  6  7  5  7  8    5  2  3  4  3  1  1  3  3  3
 345       1   3  12  25  63  96 169 189 262      1   29    8  6  7  7  5  6  5  7  7    4  2  4  2  5  2  3  3  1  2
 346       1   3  12  42  67  92 175 207 237      1   29    8  7  7  7  7  5  6  5  6    2  2  2  2  5  3  2  2  3  5
 347       1   2   5  23  80 129 148 216 244      2   29    6  5  5  7  7  7  7  7  7    5  5  3  7  2  2  1  1  1  1
 348       1   2   3  15  58 131 165 218 247      1   29    6  7  7  7  5  7  7  5  7    4  2  4  5  3  1  1  3  3  2
 349       1   3  10  19  67 129 169 183 254      1   29    8  6  5  8  7  6  6  6  6    5  2  4  3  2  1  4  2  2  3
 350       1   3  11  15  61  98 177 205 262      1   29    8  5  7  6  6  6  6  7  7    3  5  4  1  1  3  0  4  4  3
 351       1   3  10  14  60 125 177 185 261      1   29    8  6  6  6  7  7  5  6  7    4  4  4  3  3  0  3  2  1  4
 352       1   3  12  26  83  97 168 218 235      1   29    8  7  6  6  6  7  6  5  7    5  3  0  4  2  3  4  2  3  2
 353       1   3  11  27  84 119 142 180 261      1   29    8  7  6  7  6  6  5  6  7    3  2  4  1  1  3  3  6  2  3
 354       1   3  10  15  67 123 177 189 253      1   29    8  5  6  6  7  7  6  6  7    3  4  5  3  2  1  4  1  3  2
 355       1   3  10  19  77 131 138 202 261      2   29    8  6  5  7  7  6  6  6  7    2  3  5  5  2  2  2  3  2  2
 356       1   3  12  40  64 126 132 217 235      1   29    8  6  7  6  6  7  6  6  6    2  3  2  5  4  2  1  3  1  5
 357       1   2   8  15  82 109 168 218 253      1   29    6  6  7  7  7  6  6  6  7    3  4  3  4  3  1  1  2  6  1
 358       1   2   6  17  86 108 162 218 255      1   29    6  6  6  7  7  7  6  6  7    5  2  3  2  1  2  3  5  2  3
 359       1   3  12  25  83 124 136 217 238      1   30    8  6  7  7  7  7  8  6  4    3  2  3  1  5  1  0  3  3  3
 360       1   3  10  15  91 121 139 218 255      2   30    8  4  7  6  6  8  7  7  7    5  1  5  2  3  1  2  1  3  1
 361       1   3  10  19  77  95 178 212 253      1   30    8  7  6  8  7  4  7  7  6    3  2  3  2  3  2  4  3  1  1
 362       1   3  12  25  83 124 136 190 262      3   30    8  8  8  7  5  7  7  5  5    4  0  3  0  4  3  4  0  3  3
 363       1   3  10  19  60 113 175 180 263      3   30    8  7  5  8  7  7  5  8  5    6  1  2  3  1  2  2  3  1  3
 364       1   3  10  19  77 131 136 212 253      2   30    8  6  5  8  8  5  6  8  6    4  3  2  2  2  1  4  3  1  2
 365       1   3  10  15  79 127 139 209 263      1   30    8  6  6  6  7  8  8  6  5    4  3  5  2  2  0  2  1  2  3
 366       1   3  10  18  64 122 175 212 229      1   30    8  6  6  7  8  8  6  5  6    3  1  4  3  3  2  3  2  3  0
 367       1   3  10  19  65 121 177 193 255      2   30    8  6  5  7  6  8  7  6  7    4  0  4  3  3  0  3  2  3  2
 368       1   3  11  27  76 126 140 219 226      1   30    8  8  5  7  6  6  7  7  6    2  2  1  4  3  1  4  0  4  3
 369       1   3  10  13  72 121 178 202 238      1   30    8  5  6  7  8  6  7  6  7    6  2  3  2  2  0  3  3  2  1
 370       1   3  10  15  72 127 163 189 257      1   30    8  5  7  7  7  6  7  6  7    4  3  3  3  4  0  4  0  1  2
 371       1   3  11  18  58 128 167 219 231      1   30    8  7  7  6  7  5  7  6  7    2  4  0  4  2  1  2  4  3  2
 372       1   3  10  18  83 121 136 218 255      1   30    8  6  6  7  7  7  5  7  7    5  1  3  3  1  1  3  3  3  1
 373       1   3  12  26  67 124 176 184 235      1   30    8  7  7  6  5  6  7  7  7    1  2  3  2  3  3  4  1  2  3
 374       1   2   8  30  74  93 177 198 260      2   30    6  7  7  5  7  7  7  7  7    3  3  3  3  0  3  0  3  3  3
 375       1   3  10  32  89 118 142 179 257      2   30 .  8  6  6  6  8  8  6  6  6    3  2  3  4  2  3  2  3  0  2
 376       1   3  11  25  86 118 142 179 264      2   30    8  6  6  6  6  8  6  6  8    3  3  3  1  3  1  1  4  1  4
 377       1   3  10  31  79 100 177 202 229      1   30    8  6  6  8  7  7  6  6  6    4  1  1  4  0  4  4  2  2  2
 378       1   3  12  16  51 118 175 208 262      2   30    8  6  7  7  6  7  6  6  7    5  5  2  2  2  2  2  2  1  1
 379       1   3  11  25  91 100 167 180 255      1   30    8  6  7  6  7  7  7  6  6    3  2  2  3  4  2  1  2  4  1
 380       1   2   4  25  74 129 152 218 241      6   30    6  6  6  7  7  7  7  7  7    4  4  6  4  1  1  1  1  1  1
 381       1   2   8  25  66 111 161 182 263      3   30    6  6  7  7  7  7  6  7  7    3  2  2  6  3  2  1  1  3  1
 382       1   2   8  16  80 108 176 213 252      3   30    6  6  7  7  7  7  7  6  7    3  3  3  3  3  3  1  1  3  1
 383       1   3  10  15  60 118 178 218 236      1   31    8  6  7  6  8  8  7  5  7    3  2  2  4  4  1  2  1  1  0
 384       1   3  10  15  64 122 178 212 236      1   31    8  6  7  6  5  8  7  8  7    4  2  4  2  1  2  2  1  0  2
 385       1   3  10  27  79 126 133 218 236      1   31    8  7  7  8  7  6  7  7  5    1  3  3  4  2  2  1  2  1  1
 386       1   3  10  15  60 118 177 219 236      1   31    8  6  6  7  8  8  6  6  7    2  1  4  2  3  2  4  0  1  1
 387       1   3  11  33  74 100 177 205 228      1   31    8  6  8  7  6  7  6  7  7    2  1  1  3  1  3  3  3  2  1
 388       1   3  12  25  83 123 136 218 237      2   32    8  8  8  7  5  8  7  7  6    3  0  2  0  2  3  2  1  2  1
 389       1   3  10  27  79 128 133 218 234      2   32    8  6  7  8  7  5  8  8  7    1  2  2  2  2  1  1  2  1  2
 390       1   3  11  18  63 121 176 183 259      2   32    8  6  8  8  6  8  6  8  6    1  3  1  2  1  1  3  1  1  2
 391       1   3  11  25  81  99 172 179 262      2   32    8  7  6  7  8  8  6  7  7    2  2  1  2  1  2  1  2  1  2
 392       1   3  10  18  89 125 132 212 253      2   32    8  7  6  7  7  8  7  7  7    2  1  3  2  2  1  1  0  2  0
 393       1   3  10  18  79 127 132 209 259      1   33    8  6  7  8  8  7  7  8  7    1  1  2  2  0  2  2  0  1  1
 394       1   3  11  27  69 124 175 180 233      2   34    8  8  8  8  8  7  7  7  7    0  0  1  1  2  2  1  1  0  0
 395       1   3  11  18  48 125 172 216 253      4   34    8  7  7  8  8  7  7  8  8    0  0  1  1  1  1  1  1  1  1
 396       1   3  11  18  48 118 176 218 254     40   36    8  8  8  8  8  8  8  8  8    0  0  0  0  0  0  0  0  0  0
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10
SCHEDULES WITH CIRCUITS REORDERED FOR EASIER REFERENCE - REFER TO PREVIOUS LISTING
SCHED  HC     # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES
  1     0    0  0  0  0  0  0  0  0  0    12 12 12 12 12 12 12 12 12 36
  2    12    0  2  2  2  2  2  2  6  6     6  6  8  8  8  8  8  8 12 24
  3    12    0  2  2  2  3  3  4  4  4     6  6  8  8  8  8  8  8 12 24
  4    16    0  3  3  3  3  4  5  5  6     4  5  5  6  6  7  7  8 14 18
  5    16    0  4  4  4  4  4  4  4  4     6  6  6  6  6  6  6  6 16 16
  6    18    2  2  2  4  4  4  6  6  6     4  4  5  5  6  7  7  8 10 16
  7    18    2  2  2  5  5  5  5  5  5     5  5  5  5  5  5  8  8  8 18
  8    18    2  2  4  4  4  5  5  5  5     4  4  4  4  7  7  8  8 12 14
  9    18    2  2  4  4  4  5  5  5  5     5  5  5  5  5  5  8  8  8 18
 10    18    2  4  4  4  4  4  4  5  5     4  4  5  5  5  7  8  8 12 14
 11    18    3  3  3  3  3  3  6  6  6     6  6  6  6  6  6  6  6  6 18
 12    18    4  4  4  4  4  4  4  4  4     5  5  5  5  5  5  8  8  8 13
 13    18    4  4  4  4  4  4  4  4  4     5  5  5  5  5  5 10 10 10 12
 14    19    2  2  3  4  4  5  6  6  6     3  3  4  5  7  7  8  9 11 11
 15    19    2  3  3  4  4  5  5  6  6     2  2  5  6  6  7  8  8 11 13
 16    20    2  2  4  4  5  5  5  6  7     2  3  3  5  6  7  8  8 10 12
 17    20    2  3  3  3  5  6  6  6  6     2  2  4  4  7  7  8  8 11 11
 18    20    2  3  3  4  4  6  6  6  6     3  3  4  4  5  7  7  7 10 14
 19    20    2  3  3  4  4  6  6  6  6     3  3  4  4  6  6  5  8 12 12
 20    20    2  4  4  4  4  4  5  6  7     3  3  3  5  5  6  7  8  9 15
 21    20    3  3  4  4  4  5  5  6  6     3  4  4  4  4  7  7  7 12 12
 22    20    4  4  4  4  4  4  4  4  8     0  7  7  7  7  7  7  7  7  8
 23    20    4  4  4  4  4  4  5  5  6     4  4  4  4  6  6  6  6 10 14
 24    20    4  4  4  4  4  4  5  5  6     4  4  5  5  5  5  6  8 10 12
 25    21    2  3  4  4  4  5  6  6  8     3  4  4  4  4  5  6  7  9 14
 26    21    2  3  4  4  4  5  6  7  7     2  3  4  4  4  6  7  7 11 12
 27    21    2  3  5  5  5  5  5  6  6     3  3  3  4  5  6  7  9  9 11
 28    21    2  4  4  4  4  5  6  6  7     2  3  5  6  7  7  7  7  7  9
 29    21    2  4  4  5  5  5  5  5  7     3  3  3  4  4  6  7  8 10 12
 30    21    2  4  4  5  5  5  5  6  6     2  2  5  5  5  5  7  9  9 11
 31    21    2  4  4  5  5  5  5  6  6     2  4  5  6  6  7  7  7  8  8
 32    21    3  3  3  4  5  5  6  6  7     2  3  3  4  6  7  8  8  9 10
 33    21    3  3  4  4  4  4  6  6  8     2  3  5  6  6  6  7  7  8 10
 34    21    3  3  4  4  4  5  6  6  7     2  3  3  5  5  6  8  9  9 10
 35    21    3  3  4  4  4  5  6  6  7     2  3  5  5  5  6  6  8  9 11
 36    21    3  3  5  5  5  5  5  5  6     3  5  5  5  5  5  5  6  6 15
 37    21    3  4  4  4  4  5  5  6  7     3  3  3  4  5  6  7  9 10 10
 38    21    3  4  4  4  5  5  5  6  6     3  5  5  5  5  5  5  6  6 15
 39    21    4  4  4  4  5  5  5  5  6     3  4  4  6  6  6  7  7  8  9
 40    22    2  3  4  4  5  6  6  7  7     1  2  4  4  5  6  8  8  9  9
 41    22    2  3  4  4  5  6  6  7  7     2  2  3  4  5  6  8  8  9  9
 42    22    2  3  4  5  5  6  6  6  7     2  3  3  4. 4  7  7  8  8 10
 43    22    2  4  4  4  4  6  6  6  8     1  3  4  4  5  5  8  8  8 10
 44    22    2  4  4  4  4  6  6  7  7     3  3  4  4  4  5  5  6  8 14
 45    22    2  4  4  4  5  5  6  7  7     1  2  4  4  5  7  7  8  8 10
 46    22    2  4  4  4  5  6  6  6  7     2  4  4  5  6  6  7  7  7  8
 47    22    2  4  4  5  5  5  6  6  7     2  3  4  4  4  5  6  9  9 10
 48    22    2  4  5  5  5  5  6  6  6     2  2  2  5  5  5  6  8  9 12
 49    22    3  3  4  4  5  5  6  6  8     2  2  6  6  6  6  7  7  7  7
 50    22    3  3  4  5  5  6  6  6  6     1  3  3  4  5  6  7  8  9 10
 51    22    3  3  4  5  5  6  6  6  6     3  3  4  4  5  5  6  6  8 12
 52    22    3  4  4  4  5  5  6  6  7     2  2  3  4  5  6  7  7  9 11
 53    22    3  4  4  4  5  6  6  6  6     3  3  4  4  6  6  7  7  8  8
 54    22    3  4  4  4  5  6  6  6  6     3  3  4  5  6  6  6  6  7 10
 55    22    3  4  5  5  5  5  5  6  6     2  4  4  4  5  5  6  7  9 10
 56    22    4  4  4  4  5  5  6  6  6     2  3  5  5  5  6  6  7  8  9
 57    22    4  4  4  4  5  5  6  6  6     2  4  4  5  6  6  6  7  8  8
 58    22    4  4  4  4  5  5  6  6  6     3  3  3  4  4  5  6  6 10 12
 59    22    4  4  4  4  4  5  6  6  7     2  4  5  5  6  6  6  7  7  8
 60    22    4  4  4  4  4  6  6  6  6     3  4  4  5  5  5  6  6  7 11
 61    23    0  4  4  6  6  6  6  7  7     2  2  3  3  5  5  7  7  9  9
 62    23    2  3  4  5  6  6  6  7  7     1  2  4  4  4  5  6  8  9  9
 63    23    2  3  5  5  5  6  6  7  7     1  2  4  4  5  5  6  8  8  9
 64    23    2  3  5  5  6  6  6  6  7     1  2  3  4  5  5  8  8  8  8
 65    23    2  4  4  5  5  6  6  6  8     2  2  3  4  4  5  7  7  8 10
 66    23    2  4  4  5  5  6  6  6  8     2  2  4  4  6  6  6  6  8  8
 67    23    2  4  4  5  5  6  6  7  7     1  2  3  4  5  5  7  8  8  9
 68   •23    2  4  4  5  5  6  6  7  7     2  4  5  5  5  6  6  6  6  7
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10
SCHEDULES WITH CIRCUITS REORDERED FOR EASIER REFERENCE - REFER TO PREVIOUS LISTING
SCHED  HC     # HAMILT CIRCUITS ACROSS THE FCTRS     # 4-CIRCUITS ACROSS THE VERTICES
  69   23     2  4  5  5  5  6  6  6  7      1  3  4  4  4  5  6  7  8 10
  70   23     2  4  5  5  5  6  6  6  7      2  3  4  5  5  6  6  6  7  8
  71   23     3  3  4  4  4  7  7  7  7      1  2  3  4  4  6  7  7  8 10
  72   23     3  4  4  4  5  6  6  6  8      2  2  3  5  6  6  6  7  7  8
  73   23     3  4  4  5  5  6  6  6  7      1  2  4  4  4  5  7  7  8 10
  74   23     3  4  4  5  5  6  6  6  7      2  3  4  5  5  6  6  6  7  8
  75   23     3  4  4  5  5  6  6  6  7      3  3  3  5  5  5  6  7  7  8
  76   23     3  4  4  5  5  6  6  6  7      3  4  4  4  5  5  6  6  7  8
  77   23     3  4  5  5  5  5  5  7  7      1  2  3  4  6  6  6  7  8  9
  78   23     3  4  5  5  5  6  6  6  6      3  3  4  4  5  5  6  6  7  9
  79   23     4  4  4  4  5  5  6  6  8      1  4  5  5  5  6  6  6  7  7
  80   23     4  4  4  4  5  5  6  7  7      2  3  3  5  6  6  6  7  7  7
  81   23     4  4  4  5  5  5  5  7  7      1  1  4  5  5  6  6  7  8  9
  82   23     4  4  4  5  5  5  6  6  7      1  3  4  5  6  6  6  7  7  7
  83   23     4  4  4  5  5  5  6  6  7      1  3  5  5  5  6  6  7  7  7
  84   23     4  4  4  5  5  5  6  6  7      2  3  3  4  4  4  7  8  8  9
  85   23     4  4  4  5  5  5  6  6  7      2  4  4  4  5  5  6  7  7  8
  86   23     4  4  4  5  5  5  6  6  7      2  4  4  4  5  5  6  7  7  8
  87   23     4  4  4  5  5  6  6  6  6      2  3  4  4  4  5  5  7  8 10
  88   23     4  4  5  5  5  5  5  6  7      1  3  4  4  4  6  6  7  8  9
  89   23     4  4  5  5  5  5  6  6  6      3  3  4  4  5  6  6  7  7  7
  90   23     4  5  5  5  5  5  5  6  6      3  3  4  5  5  5  6  6  7  8
  91   24     2  4  4  4  6  6  7  7  8      2  3  3  4  5  5  5  6  6  9
  92   24     2  4  4  5  6  6  7  7  7      3  3  3  3  6  6  6  6  6  6
  93   24     2  4  4  6  6  6  6  7  7      1  4  4  4  4  5  6  6  6  8
  94   24     2  4  5  5  5  6  6  7  8      0  1  4  4  5  5  5  7  8  9
  95   24     2  4  5  5  6  6  6  7  7      1  1  3  3  5  5  5  5 10 10
  96   24     2  4  5  5  6  6  6  7  7      1  2  3  3  4  5  7  7  7  9
  97   24     2  4  5  5  6  6  6  7  7      1  3  3  4  5  6  6  6  7  7
  98   24     2  4  5  5  6  6  6  7  7      2  2  3  3  4  4  4  6 10 10
  99   24     2  4  5  5  6  6  6  7  7      2  3  3  4  4  4  5  6  6 11
 100   24     2  5  5  5  5  6  6  7  7      2  2  3  3  4  4  6  6  9  9
 101   24     2  5  5  5  6  6  6  6  7      4  4  4  4  5  5  5  5  6  6
 102   24     3  3  5  5  5  6  7  7  7      1  2  3  3  5  5  5  6  9  9
 103   24     3  3  5  5  6  6  6  6  8      1  2  3  4  4  5  5  6  8 10
 104   24     3  3  5  5  6  6  6  6  8      2  2  2  3  4  5  6  7  7 10
 105   24     3  4  4  4  5  7  7  7  7      2  2  5  5  5  5  5  5  7  7
 106   24     3  4  4  5  5  6  7  7  7      1  3  3  4  4  4  5  7  8  9
 107   24     3  4  4  5  5  6  7  7  7      2  2  2  5  5  5  6  7  7  7
 108   24     3  4  4  5  5  6  7  7  7      2  2  2  5  6  6  6  6  6  7
 109   24     3  4  4  5  6  6  6  6  8      1  3  3  4  4  5  5  6  7 10
 110   24     3  4  5  5  5  6  6  6  8      1  3  3  5  5  5  5  6  7  8
 111   24     3  4  5  5  5  6  6  6  8      1  4  4  4  5  5  6  6  6  7
 112   24     3  4  5  5  5  6  6  6  8      2  3  4  4  4  6  6  6  6  7
 113   24     3  4  5  5  5  6  6  7  7      1  1  3  4  5  5  6  7  8  8
 114   24     3  4  5  5  5  6  6  7  7      2  2  4  4  6  6  6  6  6  6
 115   24     3  4  5  5  5  6  6  7  7      2  3  3  4  4  4  5  5  9  9
 116   24     3  4  5  5  5  6  6  7  7      3  3  4  4  5  5  5  6  6  7
 117   24     3  4  5  5  6  6  6  6  7      0  3  3  3  4  5  6  7  8  9
 118   24     3  4  5  5  6  6  6  6  7      2  3  3  4  5  5  5  6  6  9
 119   24     3  4  5  5  6  6  6  6  7      2  3  4  4  5  5  6  6  6  7
 120   24     3  5  5  5  5  6  6  6  7      1  2  3  3  4  5  5  8  8  9
 121   24     4  4  4  4  5  6  6  7  8      3  3  3  4  4  5  5  6  7  8
 122   24     4  4  4  4  5  6  7  7  7      1  3  3  4  4  4  5  7  8  9
 123   24     4  4  4  4  5  6  7  7  7      2  2  2  2  4  6  6  7  8  9
 124   24     4  4  4  5  5  5  6  7  8      2  3  3  3  4  5  6  6  6 10
 125   24     4  4  4  5  5  5  7  7  7      3  3  4  4  5  5  6  6  6  6
 126   24     4  4  4  5  5  6  6  6  8      1  2  3  5  5  5  6  6  7  8
 127   24     4  4  4  5  5  6  6  6  8      1  3  4  5  5  5  6  6  6  7
 128   24     4  4  4  5  5  6  6  6  8      2  3  3  4  4  4  5  6  6 11
 129   24     4  4  4  5  5  6  6  7  7      2  4  4  4  5  5  5  5  7  7
 130   24     4  4  4  5  6  6  6  6  7      2  2  3  3  4  4  5  7  8 10
 131   24     4  4  4  5  6  6  6  6  7      2  2  3  4  4  5  6  7  7  8
 132   24     4  4  4  6  6  6  6  6  6      3  3  3  3  3  3  6  8  8  8
 133   24     4  4  5  5  5  5  6  6  8      1  2  3  4  5  5  6  6  6 10
 134   24     4  4  5  5  5  5  6  6  8      1  2  4  5  5  5  5  6  7  8
 135   24     4  4  5  5  5  5  6  7  7      2  2  3  3  3  5  5  5  8 12
 136   24     4  4  5  5  5  5  6  7  7      2  2  3  3  4  4  6  8  8  8
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES WITH CIRCUITS REORDERED FOR EASIER REFERENCE - REFER TO PREVIOUS LISTING

SCHED  HC     # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

 137   24     4   4   5   5   5   6   6   6   7     0   2   3   4   5   5   6   7   8   8
 138   24     4   4   5   5   5   6   6   6   7     1   4   4   4   5   5   5   5   7   8
 139   24     4   4   5   5   5   6   6   6   7     2   2   4   5   5   5   6   6   6   7
 140   24     4   4   5   5   5   6   6   6   7     2   3   3   4   4   4   5   6   8   9
 141   24     4   4   5   5   5   6   6   6   7     2   3   3   4   5   5   6   6   7   7
 142   24     4   4   5   5   5   6   6   6   7     2   3   4   4   5   5   5   6   7   7
 143   24     4   4   5   5   5   6   6   6   7     3   3   3   4   4   4   6   6   7   8
 144   24     4   4   5   5   6   6   6   6   6     1   3   4   5   5   5   6   6   6   7
 145   24     4   4   5   5   6   6   6   6   6     2   2   3   4   5   5   6   7   7   7
 146   24     4   4   5   5   6   6   6   6   6     2   4   4   4   4   5   5   5   6   9
 147   24     4   5   5   5   5   5   6   6   7     3   3   4   4   5   5   6   6   6   6
 148   24     4   5   5   5   5   6   6   6   6     3   3   3   3   4   4   4   6   8  10
 149   24     4   5   5   5   5   6   6   6   6     3   3   3   4   4   5   5   5   7   9
 150   24     5   5   5   5   5   5   5   5   8     4   4   4   4   4   4   4   4   8   8
 151   25     2   4   5   6   6   6   7   7   7     2   2   3   4   4   5   5   6   6   7
 152   25     2   4   5   6   6   6   7   7   7     2   3   3   3   5   5   5   6   6   6
 153   25     2   5   5   5   6   6   6   7   8     1   1   4   4   4   4   6   6   7   7
 154   25     2   5   6   6   6   6   6   6   7     2   3   3   3   4   5   5   5   5   9
 155   25     3   4   4   5   6   6   7   7   8     1   2   3   3   3   4   6   6   8   8
 156   25     3   4   4   6   6   6   6   7   8     1   2   4   4   4   5   5   6   6   7
 157   25     3   4   5   5   5   6   7   7   8     1   2   2   3   4   4   5   7   7   9
 158   25     3   4   5   5   6   6   7   7   7     0   3   3   4   5   5   5   6   6   7
 159   25     3   4   5   5   6   6   7   7   7     1   2   3   3   4   4   5   6   7   9
 160   25     3   4   5   6   6   6   6   6   8     2   2   2   4   4   5   5   6   7   7
 161   25     3   4   5   6   6   6   6   7   7     2   3   3   4   4   4   4   6   6   8
 162   25     3   5   5   6   6   6   6   6   7     2   2   3   3   4   4   4   5   7  10
 163   25     3   5   5   6   6   6   6   6   7     2   3   3   3   3   4   6   6   7   7
 164   25     4   4   4   4   5   6   7   8   8     1   1   1   2   4   5   6   7   8   9
 165   25     4   4   4   5   5   6   7   7   8     2   2   3   3   5   5   5   5   7   7
 166   25     4   4   4   5   6   6   6   7   8     1   2   2   5   5   5   5   6   6   7
 167   25     4   4   4   5   6   6   6   7   8     2   3   3   4   4   4   5   5   7   7
 168   25     4   4   4   6   6   6   6   7   7     3   3   3   3   4   5   5   5   5   8
 169   25     4   4   5   5   5   5   7   7   8     1   1   4   4   5   5   5   6   6   7
 170   25     4   4   5   5   5   6   7   7   7     0   2   4   4   5   5   6   6   6   6
 171   25     4   4   5   5   5   6   7   7   7     2   3   3   4   4   5   5   6   6   6
 172   25     4   4   5   5   6   6   6   6   8     2   2   3   3   4   4   5   7   7   7
 173   25     4   4   5   5   6   6   6   6   8     2   3   4   4   4   4   4   5   6   8
 174   25     4   4   5   5   6   6   6   7   7     1   3   4   4   4   5   5   5   6   7
 175   25     4   4   5   5   6   6   6   7   7     2   3   3   3   4   5   5   5   7   7
 176   25     4   4   5   5   6   6   6   7   7     2   3   3   4   4   4   5   6   6   7
 177   25     4   4   5   5   6   6   6   7   7     2   3   3   4   4   4   5   6   6   7
 178   25     4   4   5   5   6   6   6   7   7     2   3   3   4   4   5   5   6   6   6
 179   25     4   4   5   6   6   6   6   6   7     1   3   3   3   5   5   5   5   6   8
 180   25     4   4   5   6   6   6   6   6   7     2   3   3   4   4   5   5   5   6   7
 181   25     4   5   5   5   5   5   7   7   7     1   2   2   2   4   5   5   7   7   9
 182   25     4   5   5   5   5   6   6   6   8     2   2   3   3   4   4   6   6   7   7
 183   25     4   5   5   5   5   6   6   7   7     2   2   3   4   4   5   6   6   6   6
 184   25     4   5   5   5   6   6   6   6   7     0   3   4   4   5   5   5   6   6   6
 185   25     4   5   5   5   6   6   6   6   7     1   2   3   3   4   5   5   6   7   8
 186   25     4   5   5   5   6   6   6   6   7     2   3   3   4   4   4   5   5   7   7
 187   25     4   5   5   5   6   6   6   6   7     2   3   3   4   4   5   5   5   6   7
 188   25     4   5   5   5   6   6   6   6   7     2   3   4   4   4   4   4   5   6   8
 189   25     4   5   5   6   6   6   6   6   6     2   3   3   3   4   4   5   5   7   8
 190   25     5   5   5   5   5   6   6   6   7     1   2   3   5   5   5   5   6   6   6
 191   26     2   4   5   6   7   7   7   7   7     2   2   3   3   4   4   5   5   6   6
 192   26     2   5   5   5   7   7   7   7   7     3   3   3   3   4   4   5   5   5   5
 193   26     2   5   5   6   6   6   6   8   8     0   0   4   4   4   4   6   6   6   6
 194   26     2   5   5   6   6   7   7   7   7     1   1   3   3   5   5   5   5   6   6
 195   26     3   4   5   5   6   6   7   8   8     1   1   3   3   5   5   5   5   6   6
 196   26     3   4   5   5   6   7   7   7   8     0   2   2   3   3   4   5   6   7   8
 197   26     3   4   5   6   6   6   7   7   8     0   2   2   3   3   4   5   6   7   8
 198   26     3   4   6   6   6   6   6   7   8     1   2   4   4   4   4   4   5   5   7
 199   26     3   4   6   6   6   6   6   7   8     2   2   3   3   5   5   5   5   5   5
 200   26     3   5   5   5   6   6   6   8   8     2   2   2   3   3   4   5   6   6   7
 201   26     3   5   5   5   6   6   7   7   8     2   2   3   3   4   4   5   5   6   6
 202   26     3   5   5   6   6   6   7   7   7     1   1   4   4   4   4   5   5   6   6
 203   26     3   5   5   6   6   6   7   7   7     2   2   3   3   3   3   6   6   6   6
 204   26     3   5   6   6   6   6   6   7   7     2   2   3   3   4   4   5   5   6   6
```

```
REPRESENTATIVE SCHEDULES FRCM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10

SCHEDULES WITH CIRCUITS REORDERED FOR EASIER REFERENCE - REFER TO PREVIOUS LISTING

SCHED  HC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

 205   26    4  4  5  5  6  6  6  8  8     1  2  2  2  3  4  5  6  7  8
 206   26    4  4  5  5  6  6  7  7  8     1  2  3  4  4  4  4  5  6  7
 207   26    4  4  5  5  6  7  7  7  7     1  2  3  3  4  4  5  5  5  8
 208   26    4  4  5  6  6  6  6  7  8     0  2  3  4  4  5  5  5  6  6
 209   26    4  4  5  6  6  6  6  7  8     1  2  2  4  4  4  5  6  6  6
 210   26    4  4  5  6  6  6  6  7  8     1  3  3  4  4  4  4  5  5  7
 211   26    4  4  5  6  6  6  6  7  8     2  2  3  3  4  4  5  5  6  6
 212   26    4  4  5  6  6  6  7  7  7     2  2  3  3  4  5  5  5  5  6
 213   26    4  4  5  6  6  6  7  7  7     2  2  3  4  4  4  4  5  5  7
 214   26    4  4  6  6  6  6  6  6  8     2  3  3  4  4  4  4  5  5  6
 215   26    4  4  6  6  6  6  6  7  7     1  2  3  4  4  5  5  5  5  6
 216   26    4  4  6  6  6  6  6  7  7     1  3  3  4  4  4  5  5  5  6
 217   26    4  4  6  6  6  6  6  7  7     2  2  3  4  4  4  4  5  5  7
 218   26    4  5  5  5  5  6  6  8  8     1  2  2  3  3  3  4  7  7  8
 219   26    4  5  5  5  5  6  7  7  8     1  3  3  3  4  4  4  4  7  7
 220   26    4  5  5  5  6  6  6  7  8     0  3  3  3  4  5  5  5  6  6
 221   26    4  5  5  5  6  6  7  7  7     1  2  3  3  4  4  5  6  6  6
 222   26    4  5  5  5  6  6  7  7  7     1  2  3  3  4  5  5  5  5  7
 223   26    4  5  5  5  6  6  7  7  7     1  2  3  4  4  4  5  5  6  6
 224   26    4  5  5  5  6  6  7  7  7     1  2  3  4  4  4  5  5  6  6
 225   26    4  5  5  6  6  6  6  6  8     2  3  3  3  4  4  4  5  6  6
 226   26    4  5  5  6  6  6  6  7  7     1  2  2  3  4  5  5  5  6  7
 227   26    4  5  5  6  6  6  6  7  7     1  3  4  4  4  4  4  4  5  7
 228   26    4  5  5  6  6  6  6  7  7     1  3  4  4  4  4  4  5  5  6
 229   26    4  5  5  6  6  6  6  7  7     2  2  2  3  3  4  4  5  5 10
 230   26    4  5  5  6  6  6  6  7  7     2  2  3  4  4  4  5  5  5  6
 231   26    4  5  5  6  6  6  6  7  7     2  3  3  3  3  4  5  5  6  6
 232   26    4  5  6  6  6  6  6  6  7     1  2  2  5  5  5  5  5  5  5
 233   26    4  5  6  6  6  6  6  6  7     2  3  3  3  3  4  5  5  5  7
 234   26    4  6  6  6  6  6  6  6  6     2  2  2  2  3  3  4  4  8 10
 235   26    5  5  5  5  5  6  6  7  8     2  2  3  3  3  5  5  5  6  6
 236   26    5  5  5  5  5  6  7  7  7     1  2  2  4  4  5  5  5  6  6
 237   26    5  5  5  5  6  6  6  7  7     1  1  3  4  4  4  5  5  6  7
 238   26    5  5  5  5  6  6  6  7  7     1  3  3  3  4  4  5  5  5  7
 239   26    5  5  5  5  6  6  6  7  7     2  2  3  3  4  4  5  5  6  6
 240   26    5  5  5  5  6  6  6  7  7     3  3  3  3  3  4  4  4  6  7
 241   26    5  5  5  6  6  6  6  6  7     0  3  3  4  4  4  5  5  6  6
 242   26    5  5  5  6  6  6  6  6  7     2  2  2  4  4  5  5  5  5  6
 243   27    2  4  6  6  6  7  7  8  8     0  0  2  2  4  4  6  6  6  6
 244   27    3  4  6  6  7  7  7  7  7     0  2  3  3  4  4  4  5  5  6
 245   27    3  4  6  6  7  7  7  7  7     1  2  3  3  3  4  4  5  5  6
 246   27    3  5  5  6  6  6  7  8  8     1  2  3  3  4  4  4  5  5  5
 247   27    3  5  5  6  6  7  7  7  8     0  2  2  2  3  3  5  6  6  7
 248   27    3  5  5  6  6  7  7  7  8     1  2  3  3  3  4  4  4  6  6
 249   27    4  4  5  6  6  7  7  7  8     0  2  2  3  3  4  5  5  6  6
 250   27    4  4  5  6  6  7  7  7  8     1  1  2  3  4  4  5  5  5  6
 251   27    4  4  5  6  6  7  7  7  8     1  1  3  3  4  4  4  5  5  6
 252   27    4  4  5  6  6  7  7  7  8     2  2  2  2  3  3  4  4  6  8
 253   27    4  4  5  6  6  7  7  7  8     2  2  3  3  3  3  4  4  6  6
 254   27    4  4  6  6  6  6  6  8  8     1  1  2  2  3  3  5  6  6  7
 255   27    4  5  5  5  6  6  7  8  8     0  1  1  1  3  4  5  6  7  8
 256   27    4  5  5  5  6  6  7  8  8     1  2  2  2  2  3  4  6  7  7
 257   27    4  5  5  5  6  6  7  8  8     1  2  2  3  3  4  5  5  5  6
 258   27    4  5  5  5  6  7  7  7  8     2  2  2  3  3  3  5  5  5  6
 259   27    4  5  5  5  6  7  7  7  8     2  2  3  3  3  3  4  4  6  6
 260   27    4  5  5  6  6  6  7  7  8     1  2  2  3  3  3  4  5  6  7
 261   27    4  5  5  6  6  6  7  7  8     2  2  3  3  3  3  4  5  5  6
 262   27    4  5  5  6  6  6  7  7  8     2  2  3  4  4  4  4  4  4  5
 263   27    4  5  5  6  6  7  7  7  7     1  2  3  3  3  4  4  5  5  6
 264   27    4  5  5  6  6  7  7  7  7     1  2  3  3  4  4  4  4  4  7
 265   27    4  5  5  6  6  7  7  7  7     1  2  3  3  4  4  4  5  5  5
 266   27    4  5  6  6  6  6  6  7  8     1  1  3  3  3  4  4  5  6  6
 267   27    4  5  6  6  6  6  6  7  8     1  2  3  3  3  4  4  4  5  7
 268   27    4  5  6  6  6  6  6  7  8     1  2  3  3  3  4  4  4  5  7
 269   27    4  5  6  6  6  6  6  7  8     2  2  2  2  3  3  5  5  6  6
 270   27    4  5  6  6  6  6  6  7  8     2  2  3  3  3  3  3  5  5  7
 271   27    4  5  6  6  6  6  7  7  7     1  2  3  3  3  3  4  4  6  7
 272   27    4  5  6  6  6  6  7  7  7     2  2  2  3  3  4  4  5  5  6
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10
SCHEDULES WITH CIRCUITS REORDERED FOR EASIER REFERENCE - REFER TO PREVIOUS LISTING
SCHED  HC    # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES
 273   27    4  5  6  6  6  6  7  7  7    2  2  2  4  4  4  4  4  5  5
 274   27    5  5  5  5  5  6  7  8  8    2  2  2  3  4  4  4  5  5  5
 275   27    5  5  5  5  6  6  7  7  8    1  1  3  3  4  4  5  5  5  5
 276   27    5  5  5  5  6  7  7  7  7    1  1  1  3  3  4  4  5  6  8
 277   27    5  5  5  5  6  7  7  7  7    1  2  3  3  3  4  4  5  5  6
 278   27    5  5  5  5  6  7  7  7  7    2  3  3  3  3  4  4  4  5  5
 279   27    5  5  5  6  6  6  6  7  8    0  3  3  3  4  4  4  4  5  6
 280   27    5  5  5  6  6  6  6  7  8    1  2  2  3  4  4  4  4  6  6
 281   27    5  5  5  6  6  6  6  7  8    2  2  3  4  4  4  4  4  4  5
 282   27    5  5  5  6  6  6  7  7  7    0  3  3  3  4  4  4  4  5  6
 283   27    5  5  5  6  6  6  7  7  7    1  2  2  3  3  3  3  5  6  8
 284   27    5  5  5  6  6  6  7  7  7    1  2  2  3  4  4  4  5  5  6
 285   27    5  5  6  6  6  6  6  6  8    2  2  2  2  3  3  5  5  6  6
 286   27    5  5  6  6  6  6  6  7  7    2  2  2  3  3  4  4  5  5  6
 287   27    5  6  6  6  6  6  6  6  7    1  2  3  3  4  4  4  5  5  5
 288   27    5  6  6  6  6  6  6  6  7    2  2  3  3  3  3  3  4  5  8
 289   27    5  6  6  6  6  6  6  6  7    2  3  3  3  3  3  3  4  5  7
 290   27    6  6  6  6  6  6  6  6  7    3  3  3  3  3  3  3  3  3  9
 291   27    6  6  6  6  6  6  6  6  6    3  3  3  3  3  3  3  3  3  9
 292   28    3  5  6  6  7  7  7  7  8    1  1  3  3  4  4  4  4  4  4
 293   28    3  6  6  6  6  6  7  8  8    0  2  2  3  4  4  4  4  4  5
 294   28    4  4  5  6  7  7  7  8  8    1  1  2  3  3  3  4  4  5  6
 295   28    4  4  6  6  6  7  7  8  8    2  2  2  2  3  3  4  4  5  5
 296   28    4  4  6  6  7  7  7  7  8    1  1  2  3  3  4  4  4  5  5
 297   28    4  5  5  5  6  7  8  8  8    1  1  3  3  3  3  4  4  5  5
 298   28    4  5  5  6  6  6  8  8  8    1  2  2  3  3  3  4  4  4  6
 299   28    4  5  5  6  6  6  8  8  8    2  2  2  2  3  3  3  3  6  6
 300   28    4  5  5  6  6  7  7  8  8    2  2  2  2  3  3  4  4  4  6
 301   28    4  5  5  6  7  7  7  7  8    0  1  1  3  3  3  4  5  6  6
 302   28    4  5  5  6  7  7  7  7  8    0  1  3  3  3  3  4  5  5  5
 303   28    4  5  5  6  7  7  7  7  8    1  1  2  3  3  4  4  4  5  5
 304   28    4  5  5  6  7  7  7  7  8    1  1  2  3  3  4  4  4  5  5
 305   28    4  5  5  7  7  7  7  7  7    1  1  2  2  4  4  4  4  5  5
 306   28    4  5  6  6  6  6  7  8  8    1  2  2  3  3  3  4  4  5  5
 307   28    4  5  6  6  6  7  7  7  8    0  2  2  2  3  3  4  5  5  6
 308   28    4  5  6  6  6  7  7  7  8    1  1  2  2  2  2  5  5  6  6
 309   28    4  5  6  6  6  7  7  7  8    2  2  2  2  3  3  3  4  4  7
 310   28    4  5  6  6  6  7  7  7  8    2  2  2  2  3  4  4  4  4  5
 311   28    4  6  6  6  6  6  7  7  8    1  1  2  3  3  4  4  4  5  5
 312   28    4  6  6  6  6  6  7  7  8    1  2  3  3  3  3  4  4  4  5
 313   28    4  6  6  6  6  6  7  7  8    2  2  2  2  2  3  4  4  5  6
 314   28    4  6  6  6  6  7  7  7  7    1  1  2  2  3  4  4  4  5  6
 315   28    4  6  6  6  6  7  7  7  7    2  2  2  2  2  2  4  4  6  6
 316   28    4  6  6  6  6  7  7  7  7    2  2  2  2  3  3  4  4  5  5
 317   28    5  5  5  6  6  7  7  7  8    0  1  2  3  3  4  4  5  5  5
 318   28    5  5  5  6  6  7  7  7  8    0  2  2  3  3  3  4  4  5  6
 319   28    5  5  5  6  6  7  7  7  8    1  1  1  3  3  4  4  5  5  5
 320   28    5  5  5  6  6  7  7  7  8    1  1  2  2  4  4  4  4  5  5
 321   28    5  5  5  6  6  7  7  7  8    1  2  2  3  3  3  4  4  4  6
 322   28    5  5  5  6  6  7  7  7  8    1  2  2  3  3  4  4  4  4  5
 323   28    5  5  6  6  6  6  7  7  8    0  2  2  3  3  3  3  5  5  6
 324   28    5  5  6  6  6  6  7  7  8    0  2  2  3  3  4  4  4  5  5
 325   28    5  5  6  6  6  6  7  7  8    0  2  3  3  3  3  4  4  5  5
 326   28    5  5  6  6  6  6  7  7  8    1  2  2  3  3  3  3  3  5  7
 327   28    5  5  6  6  6  6  7  7  8    1  2  3  3  3  3  3  4  4  6
 328   28    6  6  6  6  6  6  6  6  8    0  0  4  4  4  4  4  4  4  4
 329   28    6  6  6  6  6  6  6  6  8    2  2  2  2  3  3  3  3  4  8
 330   28    6  6  6  6  6  6  6  6  8    3  3  3  3  3  3  3  3  4  4
 331   29    4  4  6  6  6  8  8  8  8    0  0  1  1  3  3  5  5  5  5
 332   29    4  4  6  7  7  7  7  8  8    1  1  1  1  2  2  4  4  6  6
 333   29    4  5  6  6  6  7  8  8  8    1  1  1  2  2  3  3  4  5  6
 334   29    4  5  6  6  7  7  7  8  8    0  2  2  2  2  2  4  4  5  5
 335   29    4  5  6  6  7  7  7  8  8    0  2  2  2  3  3  3  3  5  5
 336   29    4  5  6  7  7  7  7  7  8    0  1  3  3  3  3  3  3  4  5
 337   29    4  5  6  7  7  7  7  7  8    1  1  2  2  3  3  4  4  4  4
 338   29    4  6  6  6  6  7  7  8  8    1  2  2  2  3  3  3  3  4  5
 339   29    4  6  6  6  6  7  7  8  8    1  2  2  2  3  3  3  4  4  4
 340   29    4  6  6  6  7  7  7  7  8    0  1  2  3  3  3  3  4  4  5
```

```
REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR SCHEDULES OF ORDER 10
SCHEDULES WITH CIRCUITS REORDERED FOR EASIER REFERENCE - REFER TO PREVIOUS LISTING
SCHED  HC   # HAMILT CIRCUITS ACROSS THE FCTRS    # 4-CIRCUITS ACROSS THE VERTICES

 341   29   4  6  6  6  7  7  7  7  8     2  2  2  2  2  3  3  4  4  4
 342   29   5  5  5  7  7  7  7  7  8     0  1  2  2  3  4  4  4  4  4
 343   29   5  5  6  6  6  7  7  8  8     1  1  1  1  2  3  3  5  5  6
 344   29   5  5  6  6  6  7  7  8  8     1  1  2  3  3  3  3  3  4  5
 345   29   5  5  6  6  7  7  7  7  8     1  2  2  2  2  3  3  4  4  5
 346   29   5  5  6  6  7  7  7  7  8     2  2  2  2  2  2  3  3  5  5
 347   29   5  5  6  7  7  7  7  7  7     1  1  1  1  2  2  3  5  5  7
 348   29   5  5  6  7  7  7  7  7  7     1  1  2  2  3  3  3  4  4  5
 349   29   5  6  6  6  6  6  7  8  8     1  2  2  2  2  3  3  4  4  5
 350   29   5  6  6  6  6  7  7  7  8     0  1  1  3  3  3  4  4  4  5
 351   29   5  6  6  6  6  7  7  7  8     0  1  2  3  3  3  4  4  4  4
 352   29   5  6  6  6  6  7  7  7  8     0  2  2  2  3  3  3  4  4  5
 353   29   5  6  6  6  6  7  7  7  8     1  1  2  2  3  3  3  3  4  6
 354   29   5  6  6  6  6  7  7  7  8     1  1  2  2  3  3  3  4  4  5
 355   29   5  6  6  6  6  7  7  7  8     2  2  2  2  2  2  3  3  5  5
 356   29   6  6  6  6  6  6  7  7  8     1  1  2  2  2  3  3  4  5  5
 357   29   6  6  6  6  6  7  7  7  7     1  1  1  2  3  3  3  4  4  6
 358   29   6  6  6  6  6  7  7  7  7     1  2  2  2  2  3  3  3  5  5
 359   30   4  6  6  7  7  7  7  8  8     0  1  1  2  3  3  3  3  3  5
 360   30   4  6  6  7  7  7  7  8  8     1  1  1  1  2  2  3  3  5  5
 361   30   4  6  6  7  7  7  7  8  8     1  1  2  2  2  3  3  3  3  4
 362   30   5  5  5  7  7  7  8  8  8     0  0  0  3  3  3  3  4  4  4
 363   30   5  5  5  7  7  7  8  8  8     1  1  1  2  2  2  3  3  3  5
 364   30   5  5  6  6  6  8  8  8  8     1  1  2  2  2  2  3  3  4  4
 365   30   5  6  6  6  6  7  8  8  8     0  1  2  2  2  2  3  3  4  5
 366   30   5  6  6  6  6  7  8  8  8     0  1  2  2  3  3  3  3  3  4
 367   30   5  6  6  6  7  7  7  8  8     0  0  2  2  3  3  3  3  4  4
 368   30   5  6  6  6  7  7  7  8  8     0  1  1  2  2  3  3  4  4  4
 369   30   5  6  6  6  7  7  7  8  8     0  1  2  2  2  2  3  3  3  6
 370   30   5  6  6  7  7  7  7  7  8     0  0  1  2  3  3  3  4  4  4
 371   30   5  6  6  7  7  7  7  7  8     0  1  2  2  2  2  3  4  4  4
 372   30   5  6  6  7  7  7  7  7  8     1  1  1  1  3  3  3  3  3  5
 373   30   5  6  6  7  7  7  7  7  8     1  1  2  2  2  3  3  3  3  4
 374   30   5  6  7  7  7  7  7  7  7     0  0  3  3  3  3  3  3  3  3
 375   30   6  6  6  6  6  6  8  8  8     0  2  2  2  2  3  3  3  3  4
 376   30   6  6  6  6  6  6  8  8  8     1  1  1  1  3  3  3  3  4  4
 377   30   6  6  6  6  6  7  7  8  8     0  1  1  2  2  2  4  4  4  4
 378   30   6  6  6  6  7  7  7  7  8     1  1  2  2  2  2  2  2  5  5
 379   30   6  6  6  6  7  7  7  7  8     1  1  2  2  2  2  3  3  4  4
 380   30   6  6  6  7  7  7  7  7  7     1  1  1  1  1  1  4  4  4  6
 381   30   6  6  6  7  7  7  7  7  7     1  1  1  2  2  2  3  3  3  6
 382   30   6  6  6  7  7  7  7  7  7     1  1  1  3  3  3  3  3  3  3
 383   31   5  6  6  7  7  7  8  8  8     0  1  1  1  2  2  2  3  4  4
 384   31   5  6  6  7  7  7  8  8  8     0  1  1  2  2  2  2  2  4  4
 385   31   5  6  7  7  7  7  7  8  8     1  1  1  1  2  2  2  3  3  4
 386   31   6  6  6  6  7  7  8  8  8     0  1  1  1  2  2  2  3  4  4
 387   31   6  6  6  7  7  7  7  8  8     1  1  1  1  2  2  3  3  3  3
 388   32   5  6  7  7  7  8  8  8  8     0  0  1  1  2  2  2  2  3  3
 389   32   5  6  7  7  7  8  8  8  8     1  1  1  1  2  2  2  2  2  2
 390   32   6  6  6  6  8  8  8  8  8     1  1  1  1  1  1  2  2  3  3
 391   32   6  6  7  7  7  7  8  8  8     1  1  1  1  2  2  2  2  2  2
 392   32   6  7  7  7  7  7  7  8  8     0  0  1  1  2  2  2  2  3  3
 393   33   6  7  7  7  7  8  8  8  8     0  0  1  1  1  1  2  2  2  2
 394   34   7  7  7  7  8  8  8  8  8     0  0  0  0  1  1  1  1  2  2
 395   34   7  7  7  7  8  8  8  8  8     0  0  1  1  1  1  1  1  1  1
 396   36   8  8  8  8  8  8  8  8  8     0  0  0  0  0  0  0  0  0  0
```

APPENDIX B

Computer Program 1

The program (SCHPGM1) finds and lists all schedules with 2n teams which include the fixed 1-factor X_1 . It tests all the schedules for isomorphism and lists the members in each equivalence class.

The program reads input cards with the following information:

1. N = number of vertices (corresponds to 2n in the thesis body).
2. MAXSCH = maximum number of schedules to be generated and stored.

The program generates and lists all 1-factors in the routine FØRMFAC and stores them in the array FACTØR. It then forms schedules from these 1-factors in the routine FØRMSCH. The schedules are stored in the array SCHED. The routine TRANFAC aids in the equivalence testing of the schedules. It's purpose is to form an array (FTRANS) of all the mappings which can take place on the 1-factors. The actual isomorphism tests are done in the routine EQUIV. It calls two subroutines, FINDSCH and PRINTEQ, to find the equivalent schedules in the array SCHED and to print the listings of the equivalence classes.

The program was run on the 512K IBM 370-145 at the University of Victoria. It was compiled using the 0S/360 PL/1 F compiler. Compile time was 34.8 seconds. Core storage was allocated dynamically according to the values of N and MAXSCH read in. For N = 8 and MAXSCH = 500 the storage used was 128K and execution time was 2 minutes.

```
 SCHPGM1:  PROCEDURE OPTIONS (MAIN);

   /*    THIS PGM WILL GENERATE ALL 1-FACTORS FOR THE COMPLETE GRAPH WITH
         N VERTICES.  IT THEN FORMS ALL ROUND ROBIN SCHEDULES WHICH INCLUDE
         FACTOR 1.  IT ALSO FORMS EQUIVALENCE CLASSES AND PRINTS THEM OUT.   */

        DECLARE FACTOR (NFAC,N) FIXED BINARY CONTROLLED;
        DECLARE FTRANS (NFAC,NEDGE) FIXED BINARY CONTROLLED;
        DECLARE NSCH (NM2) FIXED BINARY CONTROLLED;
        DECLARE SCHED (MAXSCH,N) FIXED BINARY CONTROLLED;

        GET LIST (N);
        GET LIST (MAXSCH);
        NM1 = N-1;
        NM2 = N-2;
        NEF = 1;
        NC = 1;
  SS1:  NC = NC + 2;
        IF NC > N-2 THEN GO TO SS2;
        NEF = NEF*NC;
        GO TO SS1;
  SS2:  NFAC = (N-1)*NEF;
        NEDGE = N*(N-1)/2;
        NSS = 1;
        NSF = 500;
        ND2 = N/2;

        CALL FORMFAC;
        PUT SKIP(1) EDIT ('FACTOR','EDGES') (A,X(12),A);
        PUT SKIP(1);
        DO K = 1 TO NFAC;
           PUT SKIP(1) EDIT (K,(FACTOR(K,J) DO J = 1 TO N))
                      (F(3),9 (X(3),F(3),F(3)));
        END;

  SS3:  PUT PAGE EDIT ('LISTING OF SCHEDULES AND FACTOR NUMBERS FOR ORDER',N)
              (A,F(3));
        PUT SKIP(1);
        CALL FORMSCH;
        DO I = 1 TO NCNT;
           PUT SKIP(1) EDIT (I,'1',(SCHED(I,I1) DO I1 = 1 TO NM2))
                      (F(5),X(10),A,17 F(5));
        END;

        CALL TRANFAC;

        FREE FACTOR;

        IF N>8 THEN GO TO SS4;
        PUT PAGE EDIT ('FACTOR','TRANSFORMATIONS') (A,X(10),A);
        PUT SKIP(2) EDIT (((I,'-',J DO J = I+1 TO N) DO I = 1 TO NM1))
                    (X(4),28 (F(1),A,F(1),X(1)));
        PUT SKIP(1);
        DO K = 1 TO NFAC;
           PUT SKIP(1) EDIT (K,(FTRANS(K,IJ) DO IJ = 1 TO NEDGE))
                      (F(3),28 (X(1),F(3)));
        END;

  SS4:  CALL EQUIV;

        PUT SKIP (1);

        PUT SKIP(1) EDIT ((I2,SCHED(I2,NM1) DO I2 = 1 TO NCNT))
                   (COL(1),10 (F(4),F(5),X(3)));

   /*   *****************************************************************   */

 FORMFAC:  PROCEDURE;

   /*  TO GENERATE ALL THE FACTORS FOR N TEAMS    */

        DECLARE TAB(NTAB) FIXED BINARY CONTROLLED;
        DECLARE (TE(ND2),TE(ND2),TP(ND2)) FIXED BINARY CONTROLLED;

        ALLOCATE FACTOR(NFAC,N);
```

```
       N1 = N;
       L = 1;
       NTAB = 0;
 A1:   IF N1 = 0 THEN GO TO A2;
       NTAB = NTAB + N1;
       N1 = N1 - 2;
       GO TO A1;

 A2:   ALLOCATE TAB(NTAB),TB(ND2),TE(ND2),TP(ND2);

       DO I = 1 TO N;
          TAB(I) = I;
       END;
       N1 = N;
       N3 = 1;
       DO I = 1 TO ND2;
          TB(I) = N3;
          TP(I) = N3;
          N3 = N3 + N1;
          TE(I) = N3 - 1;
          N1 = N1 - 2;
       END;

       I = 1;
 A5:   TP(I) = TP(I) + 1;
       IF TP(I) > TE(I) THEN DO;
          IF I = 1 THEN GO TO A20;
          TP(I) = TE(I);
          I = I - 1;
          GO TO A5;
       END;
       IF I = ND2 THEN GO TO A7;
       K = TB(I+1);
       DO J = TB(I) + 1 TO TE(I);
          IF J = TP(I) THEN GO TO A6;
          TAB(K) = TAB(J);
          K = K + 1;
 A6:   END;
       I = I + 1;
       GO TO A5;

 A7:   DO I = 1 TO ND2;
          FACTOR(L,2*I-1) = TAB(TB(I));
          FACTOR(L,2*I) = TAB(TP(I));
       END;
       L = L + 1;
       I = ND2;
       GO TO A5;

 A20: FREE TAB,TB,TE,TP;
   END FORMFAC;


   /*   ***********************************************************************   */

TRANFAC:  PROCEDURE;

   /*  ROUTINE TO FIND MAPPINGS OF ALL FACTORS BY RELABELING VERTICES   */

      DECLARE TEST(N) FIXED BINARY CONTROLLED;
      DECLARE SAVE FIXED BINARY;
      DECLARE X FIXED (3);

      ALLOCATE FTRANS(NFAC,NEDGE),TEST(N);

      FTRANS = 0;
      IJ = 1;
      DO I = 1 TO NM1;
         DO J = I+1 TO N;
            DO K = 1 TO NFAC;
               IF FTRANS(K,IJ) > 0 THEN GO TO T11;
               TEST = FACTOR(K,*);
               DO L = 1 TO N;
                  IF TEST(L) = I THEN M1 = L;
                  IF TEST(L) = J THEN DO;
                     M2 = L;
```

```
                    GO TO T1;
                 END;
              END;
T1:           TEST(M1) = J;
              TEST(M2) = I;

              I1 = 1;
T2:           IF TEST(I1) < TEST(I1+1) THEN GO TO T3;
              SAVE = TEST(I1);
              TEST(I1) = TEST(I1+1);
              TEST(I1+1) = SAVE;
T3:           IF I1 = NM1 THEN GO TO T4;
              I1 = I1 + 2;
              GO TO T2;
T4:           I1 = 1;
T5:           DO I2 = I1+2 TO NM1 BY 2;
                 IF TEST(I1) < TEST(I2) THEN GO TO T6;
                 SAVE = TEST(I1);
                 TEST(I1) = TEST(I2);
                 TEST(I2) = SAVE;
                 SAVE = TEST(I1+1);
                 TEST(I1+1) = TEST(I2+1);
                 TEST(I2+1) = SAVE;
T6:           END;
              IF I1 = N-3 THEN GO TO T7;
              I1 = I1 + 2;
              GO TO T5;

T7:           I6 = (TEST(2)-2)*NEF;
              I7 = (TEST(2)-1)*NEF;
              I4 = 2;
              I3 = NEF;
T8:           X = I3;
              I3 = CEIL(X/2);
              I6 = I6+I3*(-1)**I4;
              IF I6 > I7 THEN I6 = I7;
              DO I5 = 4 TO N;
                 IF FACTOR(I6,I5) < TEST(I5) THEN DO;
                    I4 = 2;
                    GO TO T10;
                 END;
                 IF FACTOR(I6,I5) > TEST(I5) THEN DO;
                    I4 = 1;
                    GO TO T10;
                 END;
              END;
              FTRANS(I6,IJ) = K;
              FTRANS(K,IJ) = I6;
              GO TO T11;
T10:          IF I3 > 1 THEN GO TO T8;
              PUT PAGE EDIT ('ERROR1') (A);
              GO TO T20;
T11:       END;
           IJ = IJ + 1;
        END;
     END;

T20:  END TRANFAC;

  /*    **************************************************************   */

FORMSCH:  PROCEDURE;

  /*  THIS PROCEDURE PUTS FACTORS TOGETHER TO FORM ALL SCHEDULES WHICH
      INCLUDE FACTOR 1     */

     DECLARE SCHP(NM1) FIXED BINARY CONTROLLED;
     DECLARE POINT (NEDGE) FIXED BINARY CONTROLLED;
     DECLARE SVMM(ND2) FIXED BINARY CONTROLLED;

     ALLOCATE POINT(NEDGE);
     ALLOCATE SCHP(NM1),SVMM(ND2-1);
     ALLOCATE SCHED(MAXSCH,N);

     POINT = 0;
```

```
          NCNT = 0;
          DO I = 1 TO NM1;
             SCHP(I) = (I-1)*NEF+1;
          END;
          DO J = 3 TO N BY 2;
             MM = 0;
             DO M5 = 1 TO FACTOR(1,J);
                MM = MM + N - M5;
             END;
             MM = MM - N + FACTOR(1,J+1);
             POINT(MM) = 1;
          END;

          I = 2;
   F5:    IF (SCHP(I)-1)/(I*NEF) < 1 THEN GO TO F7;
          IF I = 2 THEN GO TO F20;
   F6:    SCHP(I) = (I-1)*NEF+1;
          I = I - 1;
          DO J = 3 TO N BY 2;
             MM = 0;
             DO M5 = 1 TO FACTOR(SCHP(I),J);
                MM = MM + N - M5;
             END;
             MM = MM - N + FACTOR(SCHP(I),J+1);
             POINT(MM) = 0;
          END;
          SCHP(I) = SCHP(I) + 1;
          GO TO F5;

   F7:    J1 = 0;
          DO J = 3 TO N BY 2;
             J1 = J1 + 1;
             MM = 0;
             DO M5 = 1 TO FACTOR(SCHP(I),J);
                MM = MM + N - M5;
             END;
             MM = MM - N + FACTOR(SCHP(I),J+1);
             IF POINT(MM) = 1 THEN GO TO F8;
             SVMM(J1) = MM;
          END;
          IF I = NM1 THEN GO TO F9;
          DO J1 = 1 TO ND2-1;
             POINT(SVMM(J1)) = 1;
          END;
          I = I + 1;
          GO TO F7;
   F8:    SCHP(I) = SCHP(I) + 1;
          GO TO F5;
   F9:    IF NCNT = 1000 THEN GO TO F20;
          NCNT = NCNT + 1;
          DO I1 = 2 TO NM1;
             SCHED(NCNT,I1-1) = SCHP(I1);
          END;
   F10:   I = NM1;
          GO TO F6;

   F20:    END FORMSCH;


     /*    ******************************************************************   */


   EQUIV:  PROCEDURE;

       /*   ROUTINE TO FIND EQUIVALENCE CLASSES FOR ALL SCHEDULES WITH
            FIXED FACTOR 1.     */

         DECLARE TRTBL (NEF-1,5) FIXED BINARY CONTROLLED;

         ALLOCATE NSCH (NM2);

         DO I = 1 TO NCNT;
            SCHED(I,NM1) = I;
            SCHED(I,N) = 0;
         END;
         FLAG = 0;    KTREG = 0;    JEXT = 1;
```

```
   /*  THIS PART EXCHANGES VERTICES OF FACTOR 1 WITHOUT EXCHANGING EDGES. */

      JJ = 1;
      JJM = NM2;

G1:   DO I = 1 TO NCNT;
         DO K = 1 TO NM2;
            NSCH(K) = FTRANS(SCHED(I,K),JJ);
         END;

         CALL FINDSCH;
      END;

      CALL PRINTEQ;

      IF JJM = 0 THEN GO TO G6;
      JJ = JJ + 1 + 2*JJM;
      JJM = JJM - 2;
      GO TO G1;

  /*   THIS PART EXCHANGES EDGES OF FACTOR 1    */

G6:   JJ = 2;
      JJJ = JJ + NM1;
      JJJT = NM1;
      JC = 1;
      JCT = ND2-1;
      KTRFG = 1;

G7:   DO I = 1 TO NCNT;
         DO K = 1 TO NM2;
            NSCH(K) = FTRANS(FTRANS(SCHED(I,K ),JJ),JJJ);
         END;

         CALL FINDSCH;
      END;

      CALL PRINTEQ;

      IF JCT = 1 THEN GO TO G10;
      IF JC = JCT THEN GO TO G9;
      JC = JC + 1;
      JJ = JJ + 2;
      JJJ = JJJ + 2;
      GO TO G7;

G9:   JCT = JCT - 1;
      JC = 1;
      JJ = JJJ + 2;
      JJJT = JJJT - 2;
      JJJ = JJ + JJJT;
      GO TO G7;

   /*  THIS PART CREATES A FACTOR 1 FROM OTHER FACTORS.  INITIALLY IT
       CREATES A TABLE CALLED TRTBL WHICH WILL HOLD ALL TRANSFORMATIONS
       NEEDED TO CONVERT A FIRST FACTOR INTO FACTOR 1.     */

G10: ALLOCATE TRTBL (2:NEF,5);

      TRTBL = 0;
      KTRFG = 2;
      DO J = 2 TO NEF;
         DO K = NM1 + NM1 TO NEDGE;
            IF FTRANS(J,K) = 1 THEN DO;
               TRTBL(J,1) = K;
               GO TO B1;
            END;
         END;
B1:   END;

      JCTR = 1;
B15: JCT = 0;
      DO J = 2 TO NEF;
         IF TRTBL(J,1) ¬= 0 THEN GO TO B3;
         DO JJ = 2 TO NEF;
            IF TRTBL(JJ,JCTR) = 0 THEN GO TO B4;
            DO K = NM1 + NM1 TO NEDGE;
```

```
                          IF FTRANS(J,K) = JJ THEN DO;
                             TRTBL(J,1) = K;
                             J1 = 2;
 B2:                         TRTBL(J,J1) = TRTBL(JJ,J1-1);
                             IF TRTBL(JJ,J1) = 0 THEN GO TO B3;
                             IF J1 = 5 THEN DO;
                                PUT EDIT ('ERROR TRTBL') (A);
                                GO TO SSEND;
                             END;
                             J1 = J1 + 1;
                             GO TO B2;
                          END;
                       END;
 B4:            END;
                JCT = JCT + 1;
 B3:     END;

         IF JCT = 0 THEN GO TO B6;
         JCTR = JCTR + 1;
         IF JCTR ¬= 5 THEN GO TO E15;
         PUT EDIT ('ERROR2 TRTBL') (A);
         GO TO SSEND;

    /*  THIS PART WILL THEN LINK LATER FACTORS TO A FIRST FACTOR.  */

 B6:     DO JFAC = NEF + 1 TO NFAC;
            JF = (JFAC-1)/NEF;
            DO JJ = N TO 2*N-3;
               IF FTRANS(JFAC,JJ) > NEF THEN GO TO B7;
               KK = FTRANS(JFAC,JJ);
               IF KK = 1 THEN GO TO B20;
               GO TO B8;
 B7:        END;
            PUT EDIT ('ERROR3') (A);
            GO TO SSEND;

 B8:        DO I = 1 TO NCNT;
               IF SCHED(I,JF) ¬= JFAC THEN GO TO B12;
               K1 = 1;
               J1 = 1;
               J2 = FTRANS(1,JJ);
 B85:          DO J3 = 1 TO 5;
                  IF TRTBL (KK,J3) = 0 THEN GO TO B9;
                  J2 = FTRANS(J2,TRTBL(KK,J3));
               END;
 B9:           NSCH(K1) = J2;
               IF K1 = NM2 THEN GO TO B11;
               IF J1 = JF THEN J1 = J1 + 1;
               J2 = FTRANS(SCHED(I,J1),JJ);
               K1 = K1 + 1;
               J1 = J1 + 1;
               GO TO B85;

 B11:          CALL FINDSCH;
 B12:       END;

            CALL PRINTEQ;

 B20: END;


  /*   ****************************************************************   */

FINDSCH:  PROCEDURE;

   /*  FIND SCHEDULE IN TABLE AND FORM EQUIVALENCE CLASSES    */

     DECLARE SAVE FIXED BINARY;
     DECLARE X FIXED (3);

     DO K1 = 1 TO NM2-1;
        DO K2 = K1+1 TO NM2;
           IF NSCH(K1) < NSCH(K2) THEN GO TO E4;
          .SAVE = NSCH(K1);
           NSCH(K1) = NSCH(K2);
           NSCH(K2) = SAVE;
```

```
E4:     END;
     END;

     I6 = 0;
     I3 = NCNT;
     I4 = 2;

E5:  X = I3;
     I3 = CEIL(X/2);
     I6 = I6 + I3*(-1)**I4;
     IF I6 > NCNT THEN I6 = NCNT;
     IF I6 < 1 THEN I6 = 1;
     DO I5 = 1 TO NM2;
        IF SCHED(I6,I5) < NSCH(I5) THEN DO;
           I4 = 2;
           GO TO E10;
        END;
        IF SCHED(I6,I5) > NSCH(I5) THEN DO;
           I4 = 1;
           GO TO E10;
        END;
     END;

     IF SCHED(I6,NM1) = SCHED(I,NM1) THEN GO TO E12;
     FLAG = 1;
     IHEAD = SCHED(I,NM1);
     I7 = I;
E6:  IF SCHED(I7,N) = 0 THEN GO TO E7;
     I7 = SCHED(I7,N);
     GO TO E6;
E7:  SCHED(I7,N) = SCHED(I6,NM1);
     I6 = SCHED(I6,NM1);
E8:  SCHED(I6,NM1) = IHEAD;
     IF SCHED(I6,N) = 0 THEN GO TO E12;
     I6 = SCHED(I6,N);
     GO TO E8;

E10: IF X > 1 THEN GO TO E5;
     PUT PAGE EDIT ('ERROR SEARCH') (A);
     PUT SKIP(1) EDIT ((NSCH(I5) DO I5 = 1 TO NM2)) (10 F(5));
     GO TO SSEND;

E12: END FINDSCH;

  /*   **********************************************************************   */

PRINTEQ:  PROCEDURE;

   /*  PRINT EQUIVALENCE CLASSES    */

     IF FLAG = 0 THEN GO TO P20;

     PUT PAGE EDIT ('EQUIV. CLASSES AT EXCHANGE',JEXT)  (A,F(5));

     IF KTRFG = 2 THEN PUT SKIP(1) EDIT ('EXCHANGE OF FACTORS') (A);

     IF KTRFG = 0 THEN DO;
        CALL CALCLB;
        PUT SKIP(1) EDIT ('LABEL CHANGE',L1,'-',L2) (A,F(4),A,F(2));
     END;

     IF KTRFG = 1 THEN DO;
        CALL CALCLB;
        L3 = L1;
        L4 = L2;
        JJ = JJJ;
        CALL CALCLB;
        PUT SKIP(1) EDIT ('LABEL CHANGES',L3,'-',L4,L1,'-',L2)
           (A,F(4),A,F(2),X(3),F(2),A,F(2));
     END;

     KCT = 0;
     DO I1 = 1 TO NCNT;
        IF SCHED(I1,N) = 0 THEN KCT = KCT + 1;
     END;
```

```
      PUT SKIP(2) EDIT ('#CLASSES = ',KCT) (A,F(5));
      IF KCT = 1 THEN GO TO SSEND:
      FLAG = 0:
      JEXT = JEXT + 1;

CALCLB:  PROCEDURE:
      M6 = 1;
      DO M5 = 1 TO NM1:
         M6 = M6 + N - M5;
         IF M6 > JJ THEN GO TO P5:
      END:
P5:   L1 = M5;
      L2 = N + 1 - M6 + JJ;
      END:

P20: END PRINTEQ;

  END EQUIV;

SSEND:  END SCHPGM1;

  /*   ******************************************************************   */
```

Computer Program 2

The main program (SCHPGM2) finds representatives from each equivalence class of schedules with 10 teams. It will also work for schedules with 4, 6 and 8 teams.

As a representative from each new equivalence class is generated, a line is printed and a card is punched. The main purpose of the punched output is for restart purposes. The program also prints totals for each equivalence class at the end of every level break for 1-factors from W_2 and W_3 (see Lemmas 3.6, 3.7 and 3.8). Cards are punched at every level break for 1-factors from W_2. These cards are used as input to the program PUNCHUP.

The program SCHPGM2 reads input cards which contain the following information:

1. N = number of vertices (corresponds to $2n$ in the thesis body).
2. NH = partition on Hamiltonian circuits with the first 1-factor X_1 (corresponds to 2μ in $\Phi(10,\mu)$ of Lemma 3.12).
3. NØNH = an array of a maximum of 6 possible values of 2μ not to be included in the array $P_{10}(F)$ for any schedule (see Lemma 3.12).
4. NF2 = starting value for the 1-factor from W_2. If NF2 = 0 then the program assumes it is

to generate schedules starting with the first 1-factors in W_2 and W_3; i.e., it is not a restart procedure.

5. NF3 = starting value for the 1-factor from W_3.
6. NNN = estimated number of equivalence classes for this partition $\Phi(10,\mu)$.
7. NNIS = number of representative schedules to be read in and stored in PSCHED. NNIS = 0 if it is the first run for the partition.
8. PSCHED = the array where representatives from each equivalence class are stored. If NNIS = 0, then there are no cards to be read in and the program generates all members of PSCHED. If the program is being restarted, any earlier generated representatives are read in by card and stored in PSCHED.

The program generates all 1-factors in the routine TFØRMFC. The routines FACTEST and FTR4TST perform the isomorphism tests described in Lemmas 3.6, 3.7 and 3.8 and store all relevant 1-factors in the array FACSV.

The routine TFØRMSC is the main schedule-producing routine. It calls three subroutines: TCYCLCT, TISØM and PRISØM. The routine TCYCLCT counts k-circuits and forms the arrays CYCLF and CCT which correspond to $P_{10}(F)$ and $Q_4(F)$

(see Chapter 3) respectively. The routine TISØM tests for isomorphism. In particular, TISØM stores characteristic information regarding each equivalence class which can be useful for isomorphism testing. It performs the tests described in Lemma 3.9 parts i and ii. If necessary it calls the subroutine TEQUIV to perform the test from Lemma 3.9 part iii. TEQUIV maps k-circuits onto k-circuits in an order based on low-multiplicity of numbers in $P_{10}(F)$ and $Q_4(F)$. It calls three subroutines to aid it: FILMPC2, MAPCHEK, and ECHECK (which is the final isomorphism check). If a new equivalence class has been found, TISØM calls TMPCIR to find the best combinations of 1-factors to use for mappings of k-circuits for this equivalence class. TMPCIR calls two subroutines, FILMPCC and CHECKS, to help it accomplish this. The purpose of the routine PRISØM is to print and punch the representative of a new equivalence class.

The program PUNCHUP summarizes the cards punched from SCHPGM2. Output from PUNCHUP consists of one card for every equivalence class (396 cards in all for K_{10}) with the following information on it:

1. Card identification: "3".
2. The value of 2μ for the partition $\Phi(10,\mu)$.

3. The sequence number as it was generated by SCHPGM2 (within $\Phi(10,\mu)$).
4. The total number of Hamiltonian circuits.
5. The total number of members in the equivalence class with the fixed 1-factor X_1.
6. Labels for the 1-factors in the representative schedule.

The program SCHLIST reads the cards punched by PUNCHUP and produces the listing given by Appendix A.

The programs were run on the 512K IBM 370-145 at the University of Victoria. They were compiled using the ØS/360 PL/1 F compiler with full optimization (ØPT = 2). Compile time for SCHPGM2 was 80.3 seconds. Storage was allocated dynamically and varied between 102K and 164K according to the value of NNN (the estimated number of equivalence classes for the partition $\Phi(10,\mu)$) read in. Total execution time for all runs to solve the problem of the 1-factorizations of K_{10} added to 770 minutes.

```
PUNCHUP:  PROCEDURE OPTIONS (MAIN);

  /*   THIS PROGRAM TAKES CARD OUTPUT FROM SCHPGM2 AND PUNCHES SUMMARY
       CARDS.    */

      DECLARE PUNCH OUTPUT;
      DECLARE CARD1(15) FIXED DECIMAL(5);
      DECLARE CARD2(14) FIXED DECIMAL(5);
      DECLARE (PCD,TOT) FIXED DECIMAL(5);
      OPEN FILE (PUNCH) OUTPUT;
      FFG = 0;
      CARD2(1) = 3;
A1:   GET SKIP(1) EDIT ((CARD1(I) DO I = 1 TO 15)) (15 F(5));
      IF FFG = 1 THEN GO TO A2;
      FFG = 1;
      GO TO A3;
A2:   IF CARD1(4) = PCD THEN GO TO A5;
      CARD2(3) = CARD2(3) + 1;
      CARD2(5) = TOT;
      PUT FILE (PUNCH) SKIP(1) EDIT ((CARD2(J) DO J = 1 TO 14)) (4 F(5),X(5),
          10 F(5));
      IF CARD1(1) = 0 THEN GO TO A10;
      IF CARD2(2) = CARD1(2) THEN GO TO A4;
A3:   CARD2(3) = 0;
      CARD2(2) = CARD1(2);
A4:   CARD2(4) = CARD1(5);
      DO J = 7 TO 15;
         CARD2(J-1) = CARD1(J);
      END;
      TOT = 0;
      PCD = CARD1(4);
A5:   IF CARD1(3) = 110 THEN TOT = TOT + CARD1(6)*20;
      ELSE TOT = TOT + CARD1(6) * 48;
      GO TO A1;
A10:  END PUNCHUP;

    /*   *****************************************************************   */
```

```
SCHLIST:  PROCEDURE CPTIONS (MAIN):

  /*   THIS PGM WILL MAKE PROPER LISTINGS OF SCHEDULES FROM CARD INPUT.  */
      DECLARE FACTOR (1C0C,10) FIXED BINARY:
      DECLARE SCHED (400,9) FIXED BINARY:
      DECLARE HCTOT(400) FIXED BINARY:
      DECLARE AUTCM(400);
      DECLARE CYCLF(400,9) FIXED BINARY:
      DECLARE CCT(4CC,10)  FIXED BINARY:
      DECLARE WORK(10) FIXED BINARY;
      DECLARE CYCT FIXED BINARY:
      DECLARE CYC(10) FIXED BINARY:
      DECLARE FACWK(1000) FIXED BINARY;
      DECLARE DCYCLF(400,9) FIXED BINARY:
      DECLARE DCCT(4CC,10) FIXED BINARY:
      DECLARE WK(10) FIXED BINARY;

      DECLARE SYSPRINT PRINT:

      OPEN FILE (SYSPRINT) PAGESIZE (100);

      GET LIST(N);
      NM1 = N-1:
          NM2 = N-2:
      ND2 = N/2:
      NEF = 1:
      NC = 1;
SS2:  NC = NC+2:
      IF NC > N-2 THEN GO TO SS3;
      NEF = NEF*NC;
      GO TC SS2:
SS3:  NFAC = NM1*NEF:
      NTAB = 0:
      N1 = N;
SS4:  IF N1 = 0 THEN GO TO SS5;
      NTAB = NTAB + N1;
      N1 = N1-2:
      GO TO SS4:
SS5:  PERM = 1;
      DO I = 2 TO ND2:
         PERM =  PERM*I;
      END;
       PERM =  PERM*NM1*2**ND2:

      NCNT = 0;
      TA = 0:
SS10: NCNT = NCNT+1:
      GET SKIP(1) EDIT (ICD,HCTOT(NCNT),AUTOM(NCNT),(SCHED(NCNT,J) DO J = 1
              TO NM1))    (F(5),X(10),F(5),F(5),9 F(5)):
      TA = TA + AUTCM(NCNT):
      IF ICD = 3 THEN GO TO SS10:
      NCNT = NCNT-1;

      CALL FORMFAC;

      FACWK = 0:
      DO I = 1 TO NCNT:
         DO J = 1 TC NM1:
            FACWK(SCHED(I,J)) = 1;
         END:
      END:
      I1 = 0:
      DO I = 1 TC NFAC:
         IF FACWK(I) = 1 THEN DO:
            I1 = I1 + 1:
            FACWK(I) = I1:
         END;
      END;

      CALL CYCLECT:

      CALL TORDER;

      CALL PRINT:
      GO TO SS30;
```

```
SS20: PUT PAGE EDIT ('ERROR CARD',K,CYCT,HCTOT(K))  (A,3 F(5));

  /*   ***************************************************************   */

FORMFAC:  PROCEDURE;

  /*  TO GENERATE ALL THE FACTORS FOR N TEAMS   */

      DECLARE TAB(NTAB) FIXED BINARY CONTROLLED;
      DECLARE (TB(ND2),TE(ND2),TP(ND2)) FIXED BINARY CONTROLLED;

      L = 1;

A2:   ALLOCATE TAB(NTAB),TB(ND2),TE(ND2),TP(ND2);

      DO I = 1 TO N;
         TAB(I) = I;
      END;
      N1 = N;
      N3 = 1;
      DO I = 1 TO ND2;
         TB(I) = N3;
         TP(I) = N3;
         N3 = N3 + N1;
         TE(I) = N3 - 1;
         N1 = N1 - 2;
      END;

      I = 1;
A5:   TP(I) = TP(I) + 1;
      IF TP(I) > TE(I) THEN DO;
         IF I = 1 THEN GO TO A20;
         TP(I) = TB(I);
         I = I - 1;
         GO TO A5;
      END;
      IF I = ND2 THEN GO TO A7;
      K = TB(I+1);
      DO J = TB(I) + 1 TO TE(I);
         IF J = TP(I) THEN GO TO A6;
         TAB(K) = TAB(J);
         K = K + 1;
A6:   END;
      I = I + 1;
      GO TO A5;

A7:   DO I = 1 TO ND2;
         FACTOR(L,TAB(TB(I))) = TAB(TP(I));
         FACTOR(L,TAB(TP(I))) = TAB(TB(I));
      END;
      L = L + 1;
      I = ND2;
      GO TO A5;

A20: FREE TAB,TB,TE,TP;
  END FORMFAC;

  /*   ***************************************************************   */

CYCLECT:  PROCEDURE;

   /*  THIS PROCEDURE COUNTS ALL CYCLES BETWEEN ANY TWO FACTORS AND ALSO
       COUNTS THE NUMBER OF TIMES EACH FACTOR IS CONTAINED IN EACH TYPE
       OF CYCLE.   */
  /*   ROUTINE COUNTS H.C.'S OR NON-H.C'S ONLY.  */
  /*   THIS PROCEDURE ALSO ENUMERATES THE NUMBER OF 4-CYCLES INVOLVING
       EACH VERTEX.  THIS IS USEFUL FOR CHECKING ISOMORPHISM.   */

     CYCLF = 0;
     CCT  = 0;

   DO K = 1 TO NCNT;
```

```
     CYCT = 0;

     DO I = 1 TO NM2;

        DO J = I+1 TO NM1;

           WK = 0;
           I10 = 1;
           JFG = 0;
C1:        I1 = I10;
           ICT = 0;
C2:        I2 = FACTOR(SCHED(K,I),I1);
           ICT = ICT + 1;
           WK(I1) = 1;
           WK(I2) = 1;
           CYC(ICT) = I1;
           I1 = FACTOR(SCHED(K,J),I2);
           ICT = ICT + 1;
           CYC(ICT) = I2;
           IF I1 = I10 THEN GO TO C3;
           GO TO C2;

C3:        IF JFG = 1 THEN GO TO C5;
           IF ICT = N THEN DO;
              CYCT = CYCT + 1;
              CYCLF(K,I) = CYCLF(K,I) + 1;
              CYCLF(K,J) = CYCLF(K,J) + 1;
              GO TO C6;
           END;
           JFG = 1;

C5:        IF ICT = 6 THEN GO TO C7;
           DO I6 = 1 TO 4;
              I7 = CYC(I6);
              CCT(K,I7) = CCT(K,I7) + 1;
           END;

C7:        DO I3 = 1 TO N;
              IF WK(I3) = 0 THEN DO;
                 I10 = I3;
                 GO TO C1;
              END;
           END;

C6:     END;

C10: END;

  IF CYCT ¬= HCTOT(K) THEN GO TO SS20;

  END;

 END CYCLECT;

  /*    ****************************************************************    */

TORDER:  PROCEDURE;

   /*  ORDERS H-CIRCUITS ACROSS THE FACTORS AND 4-CYCLES ACROSS THE VERTICES*/

     DO I = 1 TO NCNT;
        DO J = 1 TO NM1;
           WK(J) = CYCLF(I,J);
        END;
        J1 = 1;
T5:     MINL = 1;
        DO J = 2 TO NM1;
           IF WK(J) < WK(MINL) THEN MINL = J;
        END;
        OCYCLF(I,J1) = WK(MINL);
        IF J1 = NM1 THEN GO TO T8;
        J1 = J1 + 1;
        WK(MINL) = 99;
        GO TO T5;
T8:  END;
```

```
     LINCT = 0;
     PUT PAGE EDIT ('REPRESENTATIVE SCHEDULES FROM EACH EQUIVALENCE CLASS FOR '
             ,'SCHEDULES OF ORDER',N)  (A,A,F(3));

     IF IFLG = 0 THEN DO;
        PUT SKIP(2) EDIT ('FACTOR LIST') (A);
        PUT SKIP(2) EDIT ('FCTR','EDGES','FCTR','EDGES')
                 (A,X(5),A,X(35),A,X(5),A);
        PUT SKIP(1);
     END;

     IF IFLG = 1 THEN DO;
        PUT SKIP(2) EDIT ('SCHEDULES BY FACTOR NUMBER')  (A);
        PUT SKIP(2) EDIT ('SCHED','*  *  *  *   FACTORS   *  *  *  *',
             'ORDER','HC','# HAMILT CIRCUITS ACROSS THE FCTRS',
             '# 4-CIRCUITS ACROSS THE VERTICES')
             (A,X(4),A,X(1),A,X(2),A,X(4),A,X(4),A);
        PUT SKIP(1);
     END;

     IF IFLG = 2 THEN DO;
        PUT SKIP(2) EDIT ('SCHEDULES WITH CIRCUITS REORDERED FOR EASIER ',
             'REFERENCE - REFER TO PREVIOUS LISTING')  (A,A);
        PUT SKIP(2) EDIT ('SCHED','HC','# HAMILT CIRCUITS ACROSS THE FCTRS',
             '# 4-CIRCUITS ACROSS THE VERTICES')
             (A,X(2),A,X(4),A,X(4),A);
        PUT SKIP(1);
     END;

  END HEAD;

   END PRINT;

SS30: END SCHLIST;

  /*    **********************************************************************  */
```

```
SCHPGM2:  PROCEDURE OPTIONS (MAIN);

  /*   THIS PROGRAM FINDS REPRESENTATIVES FROM EACH EQUIVALENCE CLASS
       OF SCHEDULES WITH 10 TEAMS.  IT WILL ALSO WORK FOR SCHEDULES WITH A
       SMALLER NUMBER OF TEAMS.  IT ALSO PERFORMS CERTAIN STEPS TOWARDS THE
       COUNTING OF THE NUMBER OF MEMBERS IN EACH EQUIVALENCE CLASS.    */

      DECLARE PUNCH OUTPUT:

      DECLARE FACSV  (NFAC.NP1) FIXED BINARY CONTROLLED:
      DECLARE NONH(6):
      DECLARE ISCMTB(2.NEF) FIXED BINARY CONTROLLED:
      DECLARE PSCHED(NNN.NM1) FIXED BINARY CONTROLLED:
      DECLARE FF(NFAC) FIXED BINARY CONTROLLED;

      OPEN FILE (PUNCH) OUTPUT;

      GET LIST (N):
      PUT EDIT ('INPUT INFORMATION') (A):
      PUT SKIP(3) EDIT ('NUMBER OF VERTICES'.N) (A.F(5)):
      IF N = 10 | N = 8 | N = 6 | N = 4 THEN GO TO SS1:
      GO TO SS10:

SS1:  NP1 = N+1:
      NM1 = N-1:
      NM2 = N-2:
      ND2 = N/2:
      N999 = 999:
      NEF = 1:
      NC = 1:
SS2:  NC = NC + 2:
      IF NC > N-2 THEN GO TO SS3:
      NEF = NEF*NC:
      GO TO SS2:
SS3:  NFAC = (N-1)*NEF:
      NTAB = 0:
      N1 = N:
SS4:  IF N1 = 0 THEN GO TO SS5:
      NTAB = NTAB + N1:
      N1 = N1 - 2:
      GO TO SS4:

SS5:  GET LIST (NH):
      PUT SKIP(2) EDIT ('PARTITION ON HAMILTONIAN CIRCUITS WITH FIRST FACTOR',
            ' OF EVERY SCHEDULE'.NH)  (A.A.F(5)):
      IF NH < 0 | NH > NM2 THEN GO TO SS10:
      IF MOD(NH.2) = 1 THEN GO TO SS10:
      GET LIST ((NONH(J) DO J = 1 TO 6)):
      PUT SKIP(2) EDIT ('HAMILTONIAN COUNTS NOT INCLUDED',
            (NONH(J) DO J = 1 TO 6)) (A.6 F(5)):
      GET LIST (NF2.NF3):
      PUT SKIP(2) EDIT ('STARTING VALUES FOR SCHEDULES') (A):
      PUT SKIP(1) EDIT ('FACTOR 2'. NF2) (A.F(5)):
      PUT SKIP(1) EDIT ('FACTOR 3'. NF3) (A.F(5)):

      GET LIST (NNN);
      PUT SKIP(2) EDIT ('ESTIMATED NUMBER OF NON-ISOMORPHIC SCHEDULES'.NNN)
            (A.F(5)):

      GET LIST (NNIS):
      PUT SKIP(2) EDIT ('NUMBER OF NON-ISOMORPHIC SCHEDULES READ IN'.NNIS)
            (A.F(5)):
      ALLOCATE PSCHED(NNN.NM1):
      DO L = 1 TO NNIS:
         GET SKIP(1) EDIT ((PSCHED(L.J) DO J = 1 TO NM1)) (X(15).9 F(5)):
         PUT SKIP(1) EDIT ((PSCHED(L.J) DO J = 1 TO NM1)) (9 F(5)):
      END:

      PUT PAGE     EDIT ('FACTOR'.'EDGES') (A.X(12).A):
      PUT SKIP(1);

      CALL TFORMFC:

      PUT PAGE EDIT ('LISTING OF SCHEDULES AND FACTOR NUMBERS FOR ORDER'.N.
            ' AND PARTITION'.NH)  (A.F(3).A.F(3)):
      PUT SKIP(1);
```

```
     CALL TFORMSC;

     PUT PAGE EDIT ('END OF JOB') (A);

     GO TO SSEND;

SS10: PUT PAGE EDIT ('INCORRECT DATA') (A);

  /*    ********************************************************************   */

TFORMFC:  PROCEDURE;

  /*  TO GENERATE ALL THE FACTORS FOR N TEAMS    */

     DECLARE TAB(NTAB) FIXED BINARY CONTROLLED;
     DECLARE (TB(ND2),TE(ND2),TP(ND2)) FIXED BINARY CONTROLLED;
     DECLARE WORK1(N) FIXED BINARY CONTROLLED;
     DECLARE WORK3(N) FIXED BINARY CONTROLLED;
     DECLARE FCTR(N) FIXED BINARY CONTROLLED;
     DECLARE WORKS(NEF,NP1) FIXED BINARY CONTROLLED;
     DECLARE WKSV(2,N) FIXED BINARY CONTROLLED;

     ALLOCATE FACSV(NFAC,NP1);
     ALLOCATE WORK1(N);
     ALLOCATE WORK3(N);
     ALLOCATE TAB(NTAB),TB(ND2),TE(ND2),TP(ND2);
     ALLOCATE FCTR(N);
     ALLOCATE WORKS (NEF,NP1);
     ALLOCATE WKSV(2,N);
     ALLOCATE ISOMTB (2,NEF);
     ALLOCATE FF(NFAC);

     LM = 0;
     LL = 0;
     MFLH = 0;
     MFLNH = 0;
     MFLG1 = 0;
     MFLG2 = 0;
     MHC = 0;
     MNHC = 0;
     IHC = 0;
     LM2 = 0;
     ISOMTB = 0;
     DO K2 = 1 TO N BY 2;
        WORK1(K2) = K2 + 1;
        WORK1(K2+1) = K2;
        FACSV(1,K2) = K2+1;
        FACSV(1,K2+1) = K2;
     END;

     DO I = 1 TO N;
        TAB(I) = I;
     END;
     N1 = N;
     N3 = 1;
     DO I = 1 TO ND2;
        TB(I) = N3;
        TP(I) = N3;
        N3 = N3 + N1;
        TE(I) = N3 - 1;
        N1 = N1 - 2;
     END;

     I = 1;
A5:  TP(I) = TP(I) + 1;
     IF TP(I) > TE(I) THEN DO;
        IF I = 1 THEN GO TO A20;
        TP(I) = TB(I);
        I = I - 1;
        GO TO A5;
     END;
     IF I = ND2 THEN GO TO A7;
     K = TB(I+1);
     DO J = TB(I) + 1 TO TE(I);
        IF J = TP(I) THEN GO TO A6;
```

```
        TAB(K) = TAB(J);
        K = K + 1;
A6:  END;
     I = I + 1;
     GO TO A5;

A7:  DO I = 1 TO ND2;
        FCTR(2*I-1) = TAB(TB(I));
        FCTR(2*I) = TAB(TP(I));
     END;
     LL = LL + 1;
     CALL FACTEST;
     I = ND2;
     GO TO A5;

A20: FREE TAB,TB,TE,TP,FCTR;
     FREE WORK1,WORK3;
     FREE WKSV;
     FREE WORKS;

  /*   ******************************************************************   */

FACTEST:  PROCEDURE;

  /*   STORES IN AN ARRAY CALLED FACSV ALL FACTORS WHICH WILL COMBINE
       WITH FACTOR 1.  DEPENDING ON INPUT DATA ONE OR BOTH OF A
       NON-HAMILTONIAN AND A HAMILTONIAN FACTOR 2 ARE PICKED.
       FINALLY FACTOR 3'S ARE CHOSEN TO BE NON-ISOMORPHIC WHEN COMBINED
       WITH FACTOR 1 AND 2.        */

     IF FCTR(2) = 2 THEN DO;
        IF LM = 1 THEN GO TO W20;
        LM = 1;
        FF(1) = 1;
        ISOMTB (1,1) = NEF;
        PUT SKIP(1) EDIT (LL,(K3 DO K3 = 1 TO N),0,NEF)
              (F(4),5 (X(3),2 F(4)));
        GO TO W20;
     END;

     DO K2 = 1 TO N BY 2;
        WORK3(FCTR(K2)) = FCTR(K2+1);
        WORK3(FCTR(K2+1)) = FCTR(K2);
      END;
     DO K2 = 1 TO N;
        IF WORK3(K2) = WORK1(K2) THEN GO TO W20;
     END;

     I1 = 2;
     ICT = 1;
W3:  I1 = WORK3(I1);
     IF I1 = 1 THEN GO TO W5;
     IF MOD(I1,2) = 0 THEN DO;
        I1 = I1-1;
        GO TO W4;
     END;
     I1 = I1+1;
W4:  ICT = ICT + 1;
     GO TO W3;

W5:  IF NH = 0 & ICT = ND2 THEN GO TO W20;
     IF NH = NM2 & ICT ¬= ND2 THEN GO TO W20;
     IF FCTR(2) = 3 THEN DO;
        IF ICT = ND2 THEN DO;
           IF MFLH = 0 THEN DO;
              MFLH = 1;
              LM = LM + 1;
              DO K3 = 1 TO N;
                 FACSV(LM,K3) = WORK3(K3);
                 WKSV(LM-1,K3) = FCTR(K3);
              END;
              FACSV(LM,NP1) = 1;
              FF(LM) = LL;
           END;
           MHC = MHC + 1;
```

```
        DO I = 1 TO LM2:
           EQU(2,I) = 0:
           DO J3 = 1 TO N;
               IF WORKS(I,J3) ¬= FACSV(J12,J3) THEN GO TO D37:
               EQU(J11,I) = 0;
               GO TO D38:
D37:       END:
           EQU(1,I) = I:
D38:    END;

D30:    WK1 = 0:
        I1 = 1:
        ICTT = 0;

D1:     I3 = 1:
        I1SV = I1;

D2:     I2 = FACSV(1,I1):
        FCYC(I3) = I2:
        WK1 (I1) = 1:
        WK1(I2) = 1:
        I3 = I3 + 1:
        I1 = FACSV(J12,I2):
        FCYC(I3) = I1:
        IF I1 = I1SV THEN GO TO D3:
        I3 = I3 + 1;
        GO TO D2:

D3:     ICTT = ICTT + I3:
        DO I4 = 1 TO N:
           MAP(I4) = I4;
        END:
        IBK = 0:
        J2 = 1:

D5:     DO J1 = 1 TO I3:
           MAP(FCYC(J1)) = FCYC(MOD(J1+J2,I3)+1):
        END:
        GO TO D8;

D14:    J2 = 1:
        IBK = 1:
D15:    J3 = I3 - 2:
        DO J1 = J2 TO J2+I3/2-1:
           J4 = FCYC(J1):
           J5 = FCYC(MOD(J1+J3,I3)+1);
           MAP(J4) = J5:
           MAP(J5) = J4;
           J3 = J3-2:
        END:

D8:     DO I = 1 TO LM2:
           IF EQU(J11,I) = 0 THEN GO TO D7:
           DO J3 = 1 TO N:
               WK2(MAP(J3)) = MAP(WORKS(I,J3)):
           END:
           IF WK2(1) ¬= 4 THEN GO TO D7;
           DO J4 = 1 TO LM2:
                   DO J5 = 2 TO NM1:
                       IF WK2(J5) ¬= WORKS(J4,J5) THEN GO TO D6;
                   END:
                   IF EQU(J11,I) = EQU(J11,J4) THEN GO TO D7:
                   IF EQU(J11,I) < EQU(J11,J4) THEN DO:
                       J7 = EQU(J11,I):
                       J8 = EQU(J11,J4):
                   END:
                   ELSE DO:
                       J7 = EQU(J11,J4):
                       J8 = EQU(J11,I):
                   END:
                   DO J6 = 1 TO LM2:
                       IF EQU(J11,J6) = J8 THEN EQU(J11,J6) = J7:
                   END:
                   GO TO D7;
D6:        END:
           PUT SKIP(2) EDIT ('ERROR IN FTR4TST') (A):
           GO TO SSEND:
```

```
 D7:  END;

      IF IBK = 0 THEN DO;
          J2 = J2 + 2;
          IF J2 = I3-1 THEN GO TO D14;
          GO TO D5;
      END;

      IF J2 = I3/2 THEN GO TO D10;
      J2 = J2 + 1;
      GO TO D15;
D10:  IF ICTT = N THEN GO TO D34;
      DO I1 = 1 TO N;
          IF WK1(I1) = 0 THEN GO TO D1;
      END;

D34:  IF NH = 0 | NH = NM2 THEN GO TO D12;
      IF J11 = 2 THEN GO TO D12;
      J12 = 3;
      J11 = 2;
      DO I = 1 TO LM2;
          DO J3 = 1 TO N;
              IF WORKS(I,J3) ¬= FACSV(J12,J3) THEN GO TO D32;
              EQU(J11,I) = 0;
              GO TO D33;
D32:      END;
          EQU(2,I) = I;
D33:  END;
      GO TO D30;

D12:  DO I = 1 TO LM2;
          IF EQU(1,I) = I | EQU(2,I) = I THEN DO;
              KCT1 = 0;
              KCT2 = 0;
              DO J = I TO LM2;
                  IF EQU(1,J) = I THEN KCT1 = KCT1 + 1;
                  IF EQU(2,J) = I THEN KCT2 = KCT2 + 1;
              END;
              LM = LM + 1;
              IF KCT1 > 0 THEN ISCMTB(1,LM) = KCT1;
              IF KCT2 > 0 THEN ISCMTB(2,LM) = KCT2;
              DO K3 = 1 TO NP1;
                  FACSV(LM,K3) = WORKS(I,K3);
              END;
              FF(LM) = FF(LMS+I);
              K5 = 1;
              DO K4 = 1 TO N;
                  IF K4 > WORKS(I,K4) THEN GO TO D13;
                  WK3(K5) = K4;
                  WK3(K5+1) = WORKS(I,K4);
                  K5 = K5+2;
D13:          END;
              PUT SKIP(1) EDIT (FF(LM),(WK3(K3) DO K3 = 1 TO N), WORKS(I,NP1),
                      KCT1,KCT2)    (F(4),9 (X(3),2 F(4)));
          END;
      END;

      END FTR4TST;

W20:  END FACTEST;
  END TFORMFC;


  /*    *************************************************************  */

TFORMSC:  PROCEDURE;

  /*  THIS PROCEDURE PUTS FACTORS TOGETHER TO FORM SCHEDULES.    */

     DECLARE SCHP(NM1) FIXED BINARY CONTROLLED;
     DECLARE CYCT FIXED BINARY;
     DECLARE CYCLF  (NM1) FIXED BINARY CONTROLLED;
     DECLARE SSCHP(N) FIXED BINARY CONTROLLED;
     DECLARE WORK(NM1,N) FIXED BINARY CONTROLLED;
     DECLARE (WK(N),CYC(N),CCT(N)) FIXED BINARY CONTROLLED;
     DECLARE PWORK (NNN,NM1,N) FIXED BINARY CONTROLLED;
```

```
      DECLARE PHAMIL(NNN) FIXED BINARY CONTROLLED;
      DECLARE PFACTR(NNN,NM1,2) FIXED BINARY CONTROLLED;
      DECLARE FSING(NM1) FIXED BINARY CONTROLLED;
      DECLARE FDOUB(NM1) FIXED BINARY CONTROLLED;
      DECLARE PVERTX(NNN,N,2) FIXED BINARY CONTROLLED;
      DECLARE VSING(N) FIXED BINARY CONTROLLED;
      DECLARE VDOUB(N) FIXED BINARY CONTROLLED;
      DECLARE CFACTR(0:N,2) FIXED BINARY CONTROLLED;
      DECLARE CVERTX(0:NP1,2) FIXED BINARY CONTROLLED;
      DECLARE MPCC(N) FIXED BINARY CONTROLLED;
      DECLARE MAPP(N) FIXED BINARY CONTROLLED;
      DECLARE WTEST(N) FIXED BINARY CONTROLLED;
      DECLARE TCYCP(N) FIXED BINARY CONTROLLED;
      DECLARE WK1(N) FIXED BINARY CONTROLLED;
      DECLARE F3TOT(NNN) FIXED BINARY CONTROLLED;
      DECLARE F2TOT(NNN) FIXED BINARY CONTROLLED;
      DECLARE PMPCC(NNN,N) FIXED BINARY CONTROLLED;
      DECLARE PCL(NNN) FIXED BINARY CONTROLLED;
      DECLARE PCFAC(NNN,2) FIXED BINARY CONTROLLED;
      DECLARE PSV1(NNN,2) FIXED BINARY CONTROLLED;
      DECLARE PSVL1(NNN,2) FIXED BINARY CONTROLLED;
      DECLARE PSV2(NNN,2) FIXED BINARY CONTROLLED;
      DECLARE PSVL2(NNN,2) FIXED BINARY CONTROLLED;
      DECLARE PSYMM(NNN) FIXED BINARY CONTROLLED;

      ALLOCATE SCHP(NM1);
      ALLOCATE CYCLF  (NM1);
      ALLOCATE SSCHP(N);
      ALLOCATE WORK(NM1,N);
      ALLOCATE WK(N),CYC(N),CCT(N);
      ALLOCATE PWORK(NNN,NM1,N), PHAMIL(NNN), PFACTR(NNN,NM1,2);
      ALLOCATE FSING(NM1),FDOUB(NM1);
      ALLOCATE PVERTX(NNN,N,2), VSING(N),VDOUB(N);
      ALLOCATE CFACTR(0:N,2), CVERTX(0:NP1,2);
      ALLOCATE MPCC(N), MAPP(N), WTEST(N);
      ALLOCATE TCYCP(N), WK1(N);
      ALLOCATE F3TOT(NNN);
      ALLOCATE F2TOT(NNN);
      ALLOCATE PMPCC(NNN,N), PCL(NNN), PCFAC(NNN,2);
      ALLOCATE PSV1(NNN,2), PSVL1(NNN,2), PSV2(NNN,2), PSVL2(NNN,2);
      ALLOCATE PSYMM(NNN);

      PHAMIL = N999;
      CFACTR(0,1) = -N999;
      CFACTR(N,1) = N999;
      CVERTX(0,1) = -N999;
      CVERTX(NP1,1) = N999;
      PSV1 = 0;
      PSV2 = 0;
      PSVL1 = 0;
      PSVL2 = 0;
      F3TOT = 0;
      F2TOT = 0;
      PSYMM = 0;
      L = 0;
      ISOMFG = 0;
      NCNT = 0;

      IF NNIS = 0 THEN GO TO F2;
      INIT = 1;
      DO L = 1 TO NNIS;
         DO LJ = 1 TO NM1;
            K2 = PSCHED(L,LJ);
            SCHP(LJ) = K2;
            DO K1 = 1 TO N;
               WORK(LJ,K1) = FACSV(K2,K1);
            END;
         END;
         CALL TCYCLCT;
         CALL TISOM;
         CALL PRISOM;
      END;

F2:   INIT = 0;
      IF NH = 0 | NH = NM2 THEN LFLG = 0;
      ELSE IF N = 10 & NH = 4 THEN LFLG = 2;
           ELSE LFLG = 1;
```

```
      IHC = 0;
      IHD = 1;
      IF NH = 2 THEN DO;
          IHC = 1;
          IHD = 0;
      END;
      SCHP(1) = 1;
      DO K1 = 1 TO N;
          WORK(1,K1) = FACSV(1,K1);
      END;
      I = 2;
      DO I1 = 2 TO LM;
          IF FACSV(I1,1) > I THEN DO;
              SCHP(I) = I1;
              SSCHP(I) = I1;
              I = I + 1;
          END;
      END;
      SSCHP(N) = LM + 1;

      IF NF2 = 0 THEN GO TO F30;
      SCHP(2) = NF2;
      SCHP(3) = NF3;

F30:  JSFG = 0;

      I = 2;
F5:   IF SCHP(I) < SSCHP(I+1) THEN GO TO F7;

      IF I = 4 THEN DO;
          FF2 = FF(SCHP(2));
          FF3 = FF(SCHP(3));
          PUT SKIP(2) EDIT ('SUBTOTALS FOR FCTR GRP',    FF2,'-',FF3,
              (L1,F3TOT(L1) DO L1 = 1 TO LSV)) (A,F(4),A,F(3),X(6),500(F(3),F(5),
              X(4)));
          MULT = ISCMTB(SCHP(2)-1,SCHP(3));
          PUT SKIP(1) EDIT ('EXT. TOTS FOR FCTR GRP',    FF2,'-',FF3,
              (L1,F3TOT(L1)*MULT DO L1 = 1 TO LSV)) (A,F(4),A,F(3),X(6),
              500(F(3),F(5),X(4)));
          PUT SKIP(2);
          DO L1 = 1 TO LSV;
              F2TOT(L1) = F2TOT(L1) + MULT*F3TOT(L1);
          END;
          F3TOT = 0;
      END;

      IF I = 3 THEN DO;
          DO L1 = 1 TO LSV;
              LCCT = 0;
              DO L2 = 1 TO NM1;
                  IF PFACTR(L1,L2,1) = NH THEN LCCT = LCCT + 1;
              END;
              F2TOT(L1) = F2TOT(L1) * NM1 / LCCT;
          END;
          FF2 = FF(SCHP(2));
          PUT SKIP(2) EDIT ('COMPLETE TOTALS FOR FACTOR GROUP',FF2)  (A,F(4));
          PUT SKIP(1) EDIT ((L1,F2TOT(L1) DO L1 = 1 TO LSV))
                  (500 (F(3),F(5),X(4)));
          PUT SKIP(2);
          DO L1 = 1 TO LSV;
              IF F2TOT(L1) = 0 THEN GO TO F55;
              PUT FILE (PUNCH) SKIP(1) EDIT (2,NH,FF2,L1,PHAMIL(L1),F2TOT(L1),
                  (FF(PSCHED(L1,J)) DO J = 1 TO NM1)) (15 F(5));
F55:      END;
          F2TOT = 0;
      END;

      IF I = 2 THEN GO TO F19;
F6:   SCHP(I) = SSCHP(I);
      I = I - 1;
      K2 = SCHP(I);
      IF LFLG = 0 THEN GO TO F37;
      IF FACSV(K2,NP1) = IHC THEN JSFG = JSFG - 1;
F37:  K2 = K2 + 1;
      SCHP(I) = K2;
      GO TO F5;
```

```
F7:   K2 = SCHP(I);
      IF I = 3 & ISCNTB(SCHP(2)-1,K2) = 0 THEN GO TO F8;

      IF LFLG = 0 THEN GO TO F38;
      IF LFLG = 1 THEN DO;
         KIHC = FACSV(K2,NP1);
         IF JSFG = 2 & KIHC = IHC THEN GO TO F8;
         IF I < NM2 THEN GO TO F38;
         IF I = NM2 & JSFG = 0 & KIHC = IHD THEN GO TO F8;
         IF I = NM1 & JSFG < 2 & KIHC = IHD THEN GO TO F8;
         GO TO F38;
      END;
      IF I < 6 THEN GO TO F38;
      KIHC = FACSV(K2,NP1);
      IF JSFG = 4 & KIHC = IHC THEN GO TO F8;
      DO J20 = 1 TO 4;
         IF I = N-J20 & JSFG < 5-J20 & KIHC = IHD THEN GO TO F8;
      END;

F38: DO K3 = 2 TO I-1;
         DO K1 = 2 TO NM2;
            IF FACSV(K2,K1) = WORK(K3,K1) THEN GO TO F8;
         END;
      END;
      DO K1 = 1 TO N;
         WORK(I,K1) = FACSV(K2,K1);
      END;
      IF LFLG = 0 THEN GO TO F39;
      IF FACSV(K2,NP1) = IHC THEN JSFG = JSFG + 1;
F39: IF I = NM1 THEN GO TO F9;
      I = I + 1;
      GO TO F7;
F8:   SCHP(I) = SCHP(I) + 1;
      GO TO F5;

F9:   CALL TCYCLCT;
      ISOMFG = 0;
      CALL TISOM;
      IF ISOMFG = 1 THEN GO TO F10;
      CALL PRISOM;

F10: I = NM1;
      IF LFLG = 0 THEN GO TO F6;
      IF FACSV(SCHP(NM1),NP1) = IHC THEN JSFG = JSFG - 1;
      GO TO F6;

F19: PUT FILE (PUNCH) SKIP(3);


  /*    **************************************************************  */


TCYCLCT:  PROCEDURE;

   /*   THIS PROCEDURE COUNTS ALL HAMILTONIAN CIRCUITS BETWEEN ANY TWO
        1-FACTORS AND ALSO COUNTS THE NUMBER OF TIMES EACH FACTOR IS CONTAINED
        IN A HAMILTONIAN CIRCUIT.    */
  /*    THIS PROCEDURE ALSO ENUMERATES THE NUMBER OF 4-CIRCUITS INVOLVING
        EACH VERTEX.  THIS IS USEFUL FOR CHECKING ISOMORPHISM.    */

     CYCT = 0;
     CYCLF = 0;
     CCT = 0;

     DO I = 1 TO NM2;

        DO J = I+1 TO NM1;

           WK = 0;
           I1C = 1;
           JFG = 0;
C1:        I1 = I1C;
           ICT = 0;
C2:        I2 = WORK(I,I1);
           ICT = ICT + 1;
           WK(I1) = 1;
           WK(I2) = 1;
```

```
            CYC(ICT) = I1:
            I1 = WCRK(J,I2);
            ICT = ICT + 1:
            CYC(ICT) = I2;
            IF I1 = I10 THEN GC TO C3;
            GO TO C2:

C3:         IF JFG = 1 THEN GO TO C5:
            IF ICT = N THEN DO:
                CYCT = CYCT + 1:
                CYCLF(I) = CYCLF(I) + 1:
                CYCLF(J) = CYCLF(J) + 1:
                GO TO C6:
            END:
            JFG = 1:

C5:         IF ICT = 6 THEN GO TO C7:
            DC I6 = 1 TO 4:
                I7 = CYC(I6):
               CCT(I7) = CCT(I7) + 1:
            END:

C7:         DO I3 = 1 TO N:
                IF WK(I3) = 0 THEN DO:
                   I1C = I3:
                   GO TO C1:
                END:
            END:

C6:       END:
          ICYC = CYCLF(I):
          JN = 1:
C8:       IF NCNH(JN) > N THEN GO TO C10:
          IF ICYC = NONH(JN) THEN GO TO F10:
          JN = JN + 1:
          GO TO C8:

C10: END:
     ICYC = CYCLF(NM1):
     JN = 1:
C12: IF NCNH(JN) > N THEN GO TO C20:
     IF ICYC = NONH(JN) THEN GO TO F10:
     JN = JN + 1:
     GO TC C12:

C20: END TCYCLCT:

  /*    ************************************************************************  */

PRISCM:  PROCEDURE:

  /*   PRINTING RCUTINE FOR SCHEDULES.   */

        PUT SKIP(2) EDIT ('*',L,'*',(FF(SCHP(I1)) DO I1 = 1 TO NM1))
           (A,F(4),X(3),A,     X(7),9 F(5)):
        PUT SKIP(1) EDIT ('CIRCT',N,' - ',CYCT,(CYCLF(91) DO I1 = 1 TO NM1))
           (X(1),A,F(4),A,F(3),X(1),9 F(5)):
        PUT SKIP(0) EDIT ((CCT(I1) DO I1 = 1 TO N))  (X(68),10 F(5)):
        PUT FILE (PUNCH) SKIP(1) EDIT (1,NH,L,(SCHP(I1) DO I1 = 1 TO NM1))
           (12 F(5)):

 END PRISCM:

  /*    ************************************************************************  */

TISOM:  PROCEDURE:

  /*   PREPARES SCHEDULE TO BE TESTED FCR ISOMORPHISM.  ALL INFO REGARDING
       NON-ISOM SCHEDULES ARE STOREC IN TABLES.   */

     I2 = 1:
H1:  ISM = 1:
     DO I1 = 2 TO NM1:
```

```
          IF CYCLF(I1) < CYCLF(ISM) THEN ISM = I1;
       END;
       CFACTR(I2,1) = CYCLF(ISM);
       CFACTR(I2,2) = ISM;
       IF I2 = NM1 THEN GO TO H2;
       I2 = I2 + 1;
       CYCLF(ISM) = N999;
       GO TO H1;
H2:    I2 = 1;
H3:    ISM = 1;
       DO I1 = 2 TO N;
          IF CCT(I1) < CCT(ISM) THEN ISM = I1;
       END;
       CVERTX(I2,1) = CCT(ISM);
       CVERTX(I2,2) = ISM;
       IF I2 = N THEN GO TO H4;
       I2 = I2 + 1;
       CCT(ISM) = N999;
       GO TO H3;
H4:    IF INIT = 1 THEN GO TO H12;
       DO L = 1 TO NNN;
          IF PHAMIL(L) = N999 THEN GO TO H10;
          IF CYCT ¬= PHAMIL(L) THEN GO TO H5;
          DO J = 1 TO NM1;
             IF CFACTR(J,1) ¬= PFACTR(L,J,1) THEN GO TO H5;
             IF CVERTX(J,1) ¬= PVERTX(L,J,1) THEN GO TO H5;
          END;
          CALL TEQUIV;
          IF ISCMFG = 0 THEN GO TO H5;
          F3TOT(L) = F3TOT(L) + 1;
          GO TO H20;
H5:    END;
       PUT SKIP(1) EDIT ('TOO MANY NON-EQUIV SCHEDULES') (A);
       GO TO SSEND;

  /*   FORM NEW ISCM. GROUP.    */

H10: DO J = 1 TO NM1;
          PSCHED(L,J) = SCHP(J);
     END;
H12: PHAMIL(L) = CYCT;
     DO J = 1 TO NM1;
          PFACTR(L,J,1) = CFACTR(J,1);
          PFACTR(L,J,2) = CFACTR(J,2);
          PVERTX(L,J,1) = CVERTX(J,1);
          PVERTX(L,J,2) = CVERTX(J,2);
     END;
     PVERTX(L,N,1) = CVERTX(N,1);
     PVERTX(L,N,2) = CVERTX(N,2);

     CALL TMPCIR;

     DO J2 = 1 TO NM1;
          DO J3 = 1 TO N;
               PWORK(L,J2,J3) = WORK(J2,J3);
          END;
     END;
     DO I2 = 1 TO NM1;
          CYCLF(CFACTR(I2,2)) = CFACTR(I2,1);
          CCT(CVERTX(I2,2)) = CVERTX(I2,1);
     END;
     CCT(CVERTX(N,2)) = CVERTX(N,1);
     LSV = L;
     IF INIT = 1 THEN GO TO H20;
     F3TOT(L) = 1;


  /*   ****************************************************************   */

TEQUIV:  PROCEDURE;

   /*  TESTS FOR EQUIVALENCE BY PREPARING ALL POSSIBLE MAPPINGS.    */

   /*  BEGINS BY PREPARING CIRCUITS FOR CURRENT SCHED IN MPCC.  */

     KKISV = PCFAC(L,1);
```

```
      KK2SV = PCFAC(L,2);
      IF PSYMM(L) = 0 THEN GO TO R2;

   /*   FOR SYMMETRIC CASES, SET LIMITS FOR OTHER FACTOR COMBINATIONS
        TO TRY.    */

      DO K60 = KK2SV+1 TO NM1;
         IF CFACTR(KK2SV.1) = CFACTR(K60.1) THEN GO TO Q30:
         K65 = K60 - 1;
         GO TO Q32:
Q30:  END:
      K65 = NM1;
Q32:  IF KK1SV = 1 & CFACTR(1,1) = CFACTR(2,1) THEN DO:
         K72 = KK1SV:
         K73 = K65-1;
      END:
      ELSE DO;
         K72 = KK1SV:
         K73 = KK1SV;
      END:
      K74 = KK2SV:
      K75 = K65;

R2:   K1 = CFACTR(KK1SV.2);
      K2 = CFACTR(KK2SV.2);

R3:   KFG = 0;
      CALL FILMPC2:
      IF KFG = 1 & PCL(L) = N THEN GO TO R5:
      IF KFG = 0 & PCL(L) = 4 THEN GO TO R20:
      GO TO R80:

   /*   CIRCUIT IS HAMILTONIAN.   */

R5:   K8 = N;
      K7 = 1:
      IF PSV1(L.1) = 0 THEN DO:
         K20 = 1:
         DO K22 = 1 TO N:
            DO KFLG = 0 TO 1:
               CALL MAPCHEK;
               CALL ECHECK;
            END:
         END;
         GO TO R80;
      END:
      K22 = PSVL1(L.1);
      K4 = CVERTX(PSV1(L.1).2);
      DO K5 = 1 TO N:
         IF MPCC(K5) ¬= K4 THEN GO TO R6:
         K20 = K5:
         GO TO R7:
R6:   END;
R7:   DO KFLG = 0 TO 1;
         CALL MAPCHEK;
         CALL ECHECK:
      END:
      IF PSV1(L.2) = 0 THEN GO TO R80;
      IF K22 = PSVL1(L.2) THEN GO TO R80:
      K22 = PSVL1(L.2);
      GO TO R7;

   /*   CIRCUITS ARE NOT HAMILTONIAN.    */

R20:  IF PSV1(L.1) > 0 THEN DO;
         K4S = PSVL1(L.1);
         K4F = K4S:
         K4 = CVERTX(PSV1(L.1).2):
R21:     DO K5 = 1 TO 4;
            IF MPCC(K5) ¬= K4 THEN GO TO R22:
            KF1 = K5:
            GO TO R25:
R22:     END:
         IF K = 8 THEN DO;
            DO K5 = 1 TO 4:
               SAVE = MPCC(K5);
               MPCC(K5) = MPCC(K5+4);
```

```
                  MPCC(K5+4) = SAVE;
              END;
              GO TO R21;
          END;
          IF PSV1(L,2) = 0 THEN GO TO R80;
          IF K4 = CVERTX(PSV1(L,2),2) THEN GO TO R80;
          K4 = CVERTX(PSV1(L,2),2);
          GO TO R21;
      END;

      ELSE DO;
          K4S = 1;
          K4F = 4;
          KF1 = 1;
      END;

R25:  IF PSV2(L,1) > 0 THEN DO;
          K6S = PSVL2(L,1);
          K6F = K6S;
          K4 = CVERTX(PSV2(L,1),2);
R27:      DO K5 = 5 TO N;
              IF MPCC(K5) ¬= K4 THEN GO TO R26;
              KF2 = K5-4;
              GO TO R28;
R26:      END;
          IF PSV2(L,2) = 0 THEN GO TO R80;
          IF K4 = CVERTX(PSV2(L,2),2) THEN GO TO R80;
          K4 = CVERTX(PSV2(L,2),2);
          GO TO R27;
      END;

      ELSE DO;
          K6S = 1;
          K6F = 6;
          IF N = 8 THEN K6F = 4;
          KF2 = 1;
      END;

R28:  KL8 = 6;
      IF N = 8 THEN KL8 = 4;

  /*   ACTUAL MAPPING AND EQUIV TESTING FOR NON-HAMILT CIRCUIT.    */

R30:  DO K34 = K4S TO K4F;
          DO KFLG1 = 0 TO 1;
              K8 = 4;
              KFLG = KFLG1;
              K20 = KF1;
              K22 = K34;
              K7 = 1;
              CALL MAPCHEK;
              K8 = KL8;
              K20 = KF2;
              K7 = 5;
              DO K22 = K6S TO K6F;
                  DO KFLG  = 0 TO 1;
                      CALL MAPCHEK;
                      CALL ECHECK;
                  END;
              END;
          END;
      END;

      IF K6S = PSVL2(L,2) THEN GO TO R35;
      IF PSV2(L,2) = 0 THEN GO TO R35;
      K6S = PSVL2(L,2);
      K6F = K6S;
      GO TO R30;

R35:  IF K4S = PSVL1(L,2) THEN GO TO R80;
      IF PSV1(L,2) = 0 THEN GO TO R80;
      K4S = PSVL1(L,2);
      K4F = K4S;
      IF PSV2(L,1) = 0 THEN GO TO R30;
      K6S = PSVL2(L,1);
      K6F = K6S;
      GO TO R30;
```

```
  /*   SCHEDULES ARE NOT ISOMORPHIC.   */

R80: ISOMFG = 0;
     IF PSYMM(L) = 0 THEN GO TO R100;

  /*   FOR SYMMETRIC SCHEDULES, TRY ANOTHER FACTOR COMBINATION.  */

     IF K74 = K75 THEN DO;
        IF K72 = K73 THEN GO TO R100;
        K72 = K72 + 1;
        K74 = K72;
     END;
     K74 = K74 + 1;
     K1 = CFACTR(K72,2);
     K2 = CFACTR(K74,2);
     GO TO R3;

  /*   SCHEDULES ARE ISOMORPHIC.   */

R90: ISOMFG = 1;

  /*   ********************************************************************   */

FILMPC2:  PROCEDURE;

  /*   FILLS THE ARRAY MPCC WITH THE CIRCUITS THAT MATCH THE CIRCUITS SAVED
       IN PMPCC.   */

     WK1 = 0;
     I1P = 1;
     K10 = 0;
T1:  I1 = I1P;
     ICTP = 0;
T2:  I2 = WORK(K1,I1);
     ICTP = ICTP + 2;
     TCYCP(ICTP-1) = I1;
     TCYCP(ICTP) = I2;
     WK1(I1) = 1;
     WK1(I2) = 1;
     I1 = WORK(K2,I2);
     IF I1 = I1P THEN GO TO T5;
     GO TO T2;
T5:  IF ICTP = N THEN GO TO T6;
     K10 = K10 + 1;
     IF N = 8 THEN DO;
        IF K10 = 1 THEN K6 = 1;
           ELSE K6 = 5;
     END;
     IF N = 10 THEN DO;
        IF ICTP = 4 THEN K6 = 1;
           ELSE K6 = 5;
     END;
     GO TO T7;
T6:  K6 = 1;
     KFG = 1;
T7:  DO K5 = K6 TO K6+ICTP-1;
        MPCC(K5) = TCYCP(K5-K6+1);
     END;
     IF KFG = 1 | K10 = 2 THEN GO TO T15;
     DO K5 = 1 TO N;
        IF WK1(K5) = 1 THEN GO TO T8;
        I1P = K5;
        GO TO T1;
T8:  END;
T15: END FILMPC2;

  /*   ********************************************************************   */

MAPCHEK:  PROCEDURE;

  /*   PREPARES THE MAPPINGS FROM THE CIRCUITS.   */

     DO K5 = 0 TO K8-1;
```

```
         K9 = KFLG * K5 * 2;
         MAPP(MPCC(MOD(K5+K20-1,K8)+K7)) = PMPCC(L,MOD(K5+K22-1-K9,K8)+K7);
      END;
   END MAPCHEK;

   /*    ****************************************************************   */

ECHECK:  PROCEDURE;

   /*    THE FINAL ISOMORPHISM CHECK.  IF ISOMORPHISM OCCURS, ROUTINE
         BRANCHES TO STATEMENT R90 OF TEQUIV.    */

      DO J = 1 TO NM2;
         DO K = 1 TO N;
            WTEST(MAPP(K)) = MAPP(WORK(J,K));
         END;
         KK = WTEST(1) - 1;
         DO K = 2 TO NM1;
            IF PWORK(L,KK,K) ¬= WTEST(K) THEN GO TO Z20;
         END;
      END;
      GO TO R90;
Z20:  END ECHECK;


R100: END TEQUIV;

   /*    ****************************************************************   */

TMPCIR:  PROCEDURE;

   /*   THIS PROCEDURE FINDS THE BEST CIRCUIT FOR FUTURE ISOM MAPPINGS AND
        SAVES ASSOCIATED INFORMATION.    */

      FSING = 0;
      FDOUB = 0;
      VSING = 0;
      VDOUB = 0;

   /*   THIS PART FINDS SINGLETON AND DOUBLETON FACTORS AND VERTICES.    */

      J1 = 0;
      DO J3 = 1 TO NM1;
         IF CFACTR(J3-1,1) ¬= CFACTR(J3,1) & CFACTR(J3,1) ¬= CFACTR(J3+1,1)
            THEN DO;
            J1 = J1 + 1;
            FSING(J1) = J3;
         END;
      END;
      J2 = 0;
      DO J3 = 1 TO NM2;
         IF CFACTR(J3-1,1) ¬= CFACTR(J3,1) & CFACTR(J3,1) = CFACTR(J3+1,1)
            & CFACTR(J3+1,1) ¬= CFACTR(J3+2,1) THEN DO;
            J2 = J2 + 2;
            FDOUB(J2-1) = J3;
            FDOUB(J2) = J3+1;
         END;
      END;
      J1 = 0;
      DO J3 = 1 TO N;
         IF CVERTX(J3-1,1) ¬= CVERTX(J3,1) & CVERTX(J3,1) ¬= CVERTX(J3+1,1)
            THEN DO;
            J1 = J1 + 1;
            VSING(J1) = J3;
         END;
      END;
      J2 = 0;
      DO J3 = 1 TO NM1;
         IF CVERTX(J3-1,1) ¬= CVERTX(J3,1) & CVERTX(J3,1) = CVERTX(J3+1,1)
            & CVERTX(J3+1,1) ¬= CVERTX(J3+2,1) THEN DO;
            J2 = J2 + 2;
            VDOUB(J2-1) = J3;
            VDOUB(J2) = J3+1;
```

```
            END;
         END;

         KFG = 0;
         MPCC = 0;

    /*    FINDS CIRCUITS BETWEEN SINGLETON FACTORS.   */

         DO J5 = 1 TO NM1;
            IF FSING(J5) = 0 THEN DO;
               J4 = J5 - 1;
               GO TO Y1;
            END;
         END;
         J4 = NM1;

  Y1:    DO J1 = 1 TO J4 - 1;
            KK1 = FSING(J1);
            K1 = CFACTR(KK1,2);
            DO J2 = J1+1 TO J4;
               KK2 = FSING(J2);
               K2 = CFACTR(KK2,2);
               CALL FILMPCC;
               IF KFG = 1 THEN GO TO Y5;
               IF KFG = 0 THEN CALL CHECKS;
            END;
         END;

    /*    FIND CIRCUITS BETWEEN DOUBLETON FACTORS.   */

         DO J5 = 1 TO NM1 BY 2;
            IF FDOUB(J5) = 0 THEN DO;
               J4 = J5 - 2;
               GO TO Y2;
            END;
         END;

  Y2:    DO J1 = 1 TO J4 BY 2;
            KK1 = FDOUB(J1);
            KK2 = FDOUB(J1+1);
            K1 = CFACTR(KK1,2);
            K2 = CFACTR(KK2,2);
            CALL FILMPCC;
            IF KFG = 1 THEN GO TO Y5;
            IF KFG = 0 THEN CALL CHECKS;
         END;

    /*    FOR SYMMETRIC CASES NOT COVERED BEFORE   */

         IF MPCC(1) = 0 THEN DO;
            IF FSING(1) = 0 THEN DO;
               KK1 = 1;
               KK2 = 2;
            END;
            ELSE DO;
               KK1 = FSING(1);
               KK2 = 1;
               IF KK1 = KK2 THEN KK2 = 2;
            END;
            K1 = CFACTR(KK1,2);
            K2 = CFACTR(KK2,2);
            PSYMM(L) = 1;
            CALL FILMPCC;
            IF KFG = 0 THEN CALL CHECKS;
         END;

    /*    FILL IN ARRAYS    */

  Y5:    DO K5 = 1 TO N;
            PMPCC(L,K5) = MPCC(K5);
         END;
         IF KFG = 1 THEN PCL(L) = N;
         ELSE PCL(L) = 4;
         PCFAC(L,1) = KK1SV;
         PCFAC(L,2) = KK2SV;
         IF KFG = 1 THEN DO;
            IF VSING(1) = 0 THEN GO TO Y7;
```

```
        PSV1(L,1) = VSING(1);
        K9 = CVERTX(VSING(1),2);
        DO K5 = 1 TO N;
           IF MPCC(K5) ¬= K9 THEN GO TO Y6;
           PSVL1(L,1) = K5;
           GO TO Y20;
Y6:     END;
Y7:     IF VDOUB(1) = 0 THEN GO TO Y20;
        PSV1(L,1) = VDOUB(1);
        PSV1(L,2) = VDOUB(2);
        K8 = CVERTX(VDOUB(1),2);
        K9 = CVERTX(VDOUB(2),2);
        DO K5 = 1 TO N;
           IF MPCC(K5) = K8 THEN PSVL1(L,1) = K5;
           IF MPCC(K5) = K9 THEN PSVL1(L,2) = K5;
        END;
        GO TO Y20;
     END;
     IF K51 > 0 THEN DO;
        PSV1(L,1) = K51S;
        PSV1(L,2) = K53S;
        PSVL1(L,1) = K51;
     END;
     ELSE IF K53 > 0 THEN DO;
        PSV1(L,1) = K53S;
        PSV1(L,2) = K54S;
        PSVL1(L,1) = K53;
        PSVL1(L,2) = K54;
     END;
     IF K52 > 0 THEN DO;
        PSV2(L,1) = K52S;
        PSV2(L,2) = K55S;
        PSVL2(L,1) = K52 - 4;
     END;
     ELSE IF K55 > 0 THEN DO;
        PSV2(L,1) = K55S;
        PSV2(L,2) = K56S;
        PSVL2(L,1) = K55 - 4;
        PSVL2(L,2) = K56 - 4;
     END;


  /*   ******************************************************************   */

FILMPCC:  PROCEDURE;

  /*   FILLS THE ARRAY MPCC WITH THE BEST CIRCUITS BETWEEN 2 FACTORS.   */

     WK1 = 0;
     I1P = 1;
     K10 = 0;
T1:  I1 = I1P;
     ICTP = 0;
T2:  I2 = WORK(K1,I1);
     ICTP = ICTP + 2;
     TCYCP(ICTP-1) = I1;
     TCYCP(ICTP) = I2;
     WK1(I1) = 1;
     WK1(I2) = 1;
     I1 = WORK(K2,I2);
     IF I1 = I1P THEN GO TO T5;
     GO TO T2;
T5:  IF ICTP = N THEN GO TO T6;
  /*   KFG = 2 MEANS THAT THIS CIRCUIT CANNOT BE BETTER THAN AN EARLER ONE */
     IF KFG = 2 THEN GO TO T15;
     K10 = K10 + 1;
     IF N = 8 THEN DO;
        IF K10 = 1 THEN K6 = 1;
           ELSE K6 = 5;
     END;
     IF N = 10 THEN DO;
        IF ICTP = 4 THEN K6 = 1;
           ELSE K6 = 5;
     END;
     GO TO T7;
T6:  K6 = 1;
```

```
      KFG = 1;
T7:   DO K5 = K6 TO K6+ICTP-1;
          MPCC(K5) = TCYCF(K5-K6+1);
      END;
      IF KFG = 1 | K10 = 2 THEN GO TO T10;
      DO K5 = 1 TO N;
          IF WK1(K5) = 1 THEN GO TO T8;
          I1P = K5;
          GO TO T1;
T8:   END;
  /*    SAVES FACTORS IN CASE THIS CIRCUIT TURNS OUT TO BE THE BEST    */
T10:  KK1SV = KK1;
      KK2SV = KK2;
T15:  END FILMPCC;


  /*    **********************************************************************  */


CHECKS:  PROCEDURE;

  /*   THIS ROUTINE CHECKS NON-HAMIL. CIRCUITS FOR SINGLETON AND DOUBLETON
       VERTICES AND STORES THEIR LOCATIONS.     */

      K51 = 0;
      K52 = 0;
      K53 = 0;
      K54 = 0;
      K55 = 0;
      K56 = 0;
      K53S = 0;
      K55S = 0;

      DO K6 = 1 TO N;
          K4 = VSING(K6);
          IF K4 = 0 THEN GO TO U6;
          K9 = CVERTX(K4,2);
          IF K51 > 0 THEN GO TO U2;
          DO K5 = 1 TO 4;
              IF MPCC(K5) ¬= K9 THEN GO TO U1;
              K51S = K4;
              K51 = K5;
              GO TO U4;
U1:       END;
U2:       IF K52 > 0 THEN GO TO U4;
          DO K5 = 5 TO N;
              IF MPCC(K5) ¬= K9 THEN GO TO U3;
              K52S = K4;
              K52 = K5;
              GO TO U4;
U3:       END;
U4:       IF K52*K51 ¬= 0 THEN GO TO U10;
      END;

U6:   DO K6 = 1 TO N BY 2;
          K3 = VDOUB(K6);
          IF K3 = 0 THEN GO TO U20;
          K4 = VDOUB(K6+1);
          K8 = CVERTX(K3,2);
          K9 = CVERTX(K4,2);
          IF K51 > 0 THEN GO TO U7;
          K53 = 0;
          K54 = 0;
          K53S = 0;
          DO K5 = 1 TO 4;
              IF MPCC(K5) = K8 THEN DO;
                  K53 = K5;
                  K53S = K3;
              END;
              IF MPCC(K5) = K9 THEN DO;
                  K54 = K5;
                  K54S = K4;
              END;
          END;
          IF K53 > 0 & K54 = 0 THEN DO;
              K51 = K53;
              K51S = K53S;
```

```
               K53S = K4;
            END;
            IF K54 > 0 & K53 = 0 THEN DO;
               K51 = K54;
               K51S = K54S;
               K53S = K3;
            END;
 U7:        IF K52 > 0 THEN GO TC U8;
            K55 = 0;
            K56 = 0;
            K55S = C;
            DC K5 = 5 TO N;
               IF MPCC(K5) = K8 THEN DC;
                  K55 = K5;
                  K55S = K3;
               END;
               IF MPCC(K5) = K9 THEN DO;
                  K56 = K5;
                  K56S = K4;
               END;
            END;
            IF K55 > 0 & K56 = 0 THEN DC;
               K52 = K55;
               K52S = K55S;
               K55S = K4;
            END;
            IF K56 > 0 & K55 = 0 THEN DO;
               K52 = K56;
               K52S = K56S;
               K55S = K3;
            END;
 U8:        IF K52*K51 ¬= 0 THEN GO TO U10;
         END;
         GO TO U20;
 U10: KFG = 2;
 U20: END CHECKS;

 Y20:  END TMPCIR;

 H20: END TISCM;

 F20:  END TFORMSC;

 SSEND:  END SCHPGM2;

   /*   **********************************************************************   */
```

VITA

Surname: GELLING Given Names: ERIC NEIL

Place of Birth: VICTORIA, B.C. Date of Birth: DEC. 28, 1945

Educational Institutions Attended, with Dates of Entering and Leaving:

UNIVERSITY OF VICTORIA, VICTORIA 1963 to 1968

UNIVERSITY OF VICTORIA, VICTORIA 1971 to 1973

Degrees, Diplomas, Etc., Awarded, with Dates and Names of Institutions:

B.A. 1968 *University of Victoria, Victoria*

Honors and Awards:

Vancouver Sun Scholarships, 1963/64 and 1964/65

Government of British Columbia Scholarships, 1963/64 to 1967/68

Percy H. Elliott Memorial Scholarship, 1963/64

University Extension Association of Victoria Scholarship 1965/66

University of Victoria Graduate Scholarship, 1972/73

Appointments with Dates and Names of Employers:

I.B.M. Canada Limited 1968 to 1971

Publications:

PARTIAL COPYRIGHT LICENSE

Title of Thesis

ON 1-FACTORIZATIONS OF THE COMPLETE
GRAPH AND THE RELATIONSHIP TO
ROUND ROBIN SCHEDULES

Author

Eric Neil Gelling
Name

July 1973
Date