JOURNAL OF COMPUTER AND SYSTEM SCIENCES 25, 332–359 (1982)

Subcomplete Generalizations of Graph Isomorphism*

CHRISTOPH M. HOFFMANN

Department of Computer Science, Purdue University, West Lafayette, Indiana 47907

Received February 28, 1981; revised March 10, 1982

We present eight group-theoretic problems in **NP** one of which is a reformulation of graph isomorphism. We give technical evidence that none of the problems is **NP**-complete, and give polynomial time reductions among the problems. There is a good possibility that seven of these problems are harder than graph isomorphism (relative to polynomial time reduction), so that they might be examples of natural problems of *intermediate* difficulty, situated properly between the class of **NP**-complete problems and the class **P** of problems decidable in deterministic polynomial time. Because of strong structural similarity, two of the apparently harder problems can be interpreted as *generalized isomorphism* and *generalized automorphism*, respectively. Whether these problems ultimately prove to be harder than graph isomorphism seems to depend, in part, on the open problem whether every permutation group of degree n arises as the automorphism group of a combinatorial structure of size polynomial in n. Finally, we give an $O(n^2 \cdot k)$ algorithm for constructing the *centralizer* of a permutation group of degree n presented by a generating set of k permutations. Note that we may assume that k is $O(n \cdot \log n)$, without loss of generality. This problem is a special case of one of the harder group-theoretic problems. From the method of constructing the centralizer, we recover results about the group-theoretic structure of the centralizer. Furthermore, applying our algorithm for intersecting with a normalizing permutation group, we show how to find the *center* of a permutation group of degree n in $O(n^6)$ steps, having constructed the centralizer of the group first.

1. INTRODUCTION

To date, the complexity status of graph isomorphism is still open. Although clearly in **NP**, it has neither been shown to be **NP**-complete, nor has there been given a deterministic polynomial time algorithm for the problem. The length of time graph isomorphism has resisted the many attacks on it suggests that it is a problem of intermediate difficulty, i.e., a problem which neither is in **P** nor is an **NP**-complete problem (assuming of course that $\mathbf{P} \neq \mathbf{NP}$).

Extensive work on graph isomorphism has resulted so far in the following infinite hierarchies of graphs possessing a polynomial time isomorphism test: graphs of bounded genus [9, 10, 16, 17, 19, 26], graphs of bounded valence [7, 21], and graphs of bounded eigenvalue multiplicities [2]. For example, isomorphism of eigenvalue

* This work has been supported in part by the National Science Foundation under Grant No. MCS-80-21066.

0022-0000/82/060332–28$02.00/0

multiplicity k may be tested in $O(n^{4k+c})$ steps, where n is the number of graph vertices and c is a suitable constant.

Much recent research on graph isomorphism has concentrated on finding problems polynomial time equivalent to graph isomorphism. Such problems are called *isomorphism complete*, and there is a sizable list of them [4, 23, 24]. Almost all known isomorphism complete problems are again isomorphism problems associated with topological, combinatorial, or algebraic structures. There are, however, at least two interesting exceptions: The problem of determining the automorphism group of a graph and the problem of counting the number of isomorphisms between two graphs [23].

The isomorphism completeness of the problem of counting the number of isomorphisms is of interest because it may be taken as technical evidence against the possibility that graph isomorphism is **NP**-complete. For **NP**-complete existence problems, the associated counting problem is believed to be more difficult [12, 29].

The isomorphism completeness of the problem of determining the automorphism group of a graph is of interest precisely because it is not an isomorphism question of some structure. We therefore view it as a different perspective on the nature of graph isomorphism and see significance in the fact that the recent development of efficient isomorphism tests for two of the above-mentioned hierarchies of graphs has been based largely on group-theoretic techniques.

This paper is divided into two parts. In the first part, we present a number of group-theoretic problems which appear to fall into two levels of difficulty relative to polynomial time reduction, with graph isomorphism on the easier level. We can interpret the more difficult problems as natural generalizations of the problems of testing isomorphism and of determining the automorphism group for graphs. Previously proposed generalizations of graph isomorphism, e.g., subgraph isomorphism [12], or fixpoint-free automorphism [20], have been shown to be **NP**-complete. In contrast, for the generalizations of this paper, we can present technical evidence against the possibility that any of our problems is **NP**-complete, and we therefore believe that they are subcomplete.

In the second part of this paper, we give polynomial time algorithms for two special cases of one of the harder problems. Specifically, given an arbitrary permutation group $G < S_n$ by a generating set of k permutations, we show how to find generators for the *centralizer* (in S_n) of G, in $O(n^2 \cdot k)$ steps. This generator set, found for the centralizer of G, has the special properties required by the $O(n^2)$ group membership test of [8], i.e., it is a strong generating set in the sense of [28]. (See also Theorem 2.5 below.) Exponential algorithms for determining generators for the centralizer of a group have previously been advocated [6, 27]. We have recently learned, however, that our centralizer algorithm was independently discovered by Fontet [31]. Applying our algorithm for intersecting with a normalizing permutation group [15], we give an $O(n^2 \cdot k + n^6)$ algorithm for determining the *center* of G. The only polynomial time algorithm previously known for this problem is due to Luks, [22], and requires a squaring of the group degree, thus requiring at least $O(n^4 \cdot k + n^{12})$ steps.

The group-theoretic problems to be described in the first part appear to fall into two levels of difficulty. Representative harder problems include the double coset membership problem and the group intersection problem. Graph isomorphism is apparently easier. The distinguishing characteristic of all problems appears to be that the existence problem and its associated counting problem are polynomial time equivalent.

Assuming that $\mathbf{P} \neq \mathbf{NP}$, it has been shown that there exist problems which are neither in **P** nor are **NP**-complete [18]. Such problems of intermediate difficulty typically have been constructed as artificial subproblems of certain **NP**-complete problems, and are known to contain hierarchies relative to polynomial time reduction.

Our problems are among the first examples of *natural* problems of unknown complexity status which might participate in a subcomplete hierarchy. Other natural problems polynomially equivalent to our harder problems have been proposed by Luks [22]. Previously, Booth has proposed the two level hierarchy of group isomorphism < graph isomorphism (here each group is given as a list of elements) [3, 24]. Thus this paper and [22] might add a new (and natural) level. It is interesting to note that there is also an **NP**-complete generalization of the harder group-theoretic problems given here which we sketch below.

Our suggestion that the problems presented differ in difficulty is, of course, quite tentative: For one, research into the complexity of these group-theoretic problems is fairly recent, and given the evidence presented here, our conjecture is not compelling. On the other hand, there has been research that has found other problems polynomially equivalent to the harder problems presented here, while also finding subcomplete problems apparently harder yet [22]; this research has not produced evidence against the properness of the two levels of difficulty. Moreover, the open problem of whether all permutation groups arise as the automorphism group of a small combinatorial structure is related and has remained open in mathematics for several decades [11].

We outline several of the problems presented. Let A and B be two permutation groups of degree n, i.e., A and B are subgroups of S_n, the symmetric group of degree n containing all permutations of the set $\{1,..., n\}$. Let $\pi \in S_n$ be any permutation. The set

$$A\pi B = \{\alpha\pi\beta \mid \alpha \in A, \beta \in B\}$$

is called the *double coset* of A and B containing π. Double cosets induce an equivalence partition on S_n. The *double coset membership problem* is the question whether two permutations lie in the same double coset, i.e., whether they are equivalent. In Section 3, we discuss why this problem is a natural generalization of graph isomorphism.

Of difficulty equal to the double coset membership problem is the *group intersection problem*: Given two permutation groups A and B of degree n, by generating sets, determine a generating set for their intersection $C = A \cap B$, again a group. In [15], we have shown that a polynomial time solution to the group intersection problem gives a polynomial time solution for graph isomorphism. In Section 3, we show that a polynomial time algorithm for the double coset membership problem gives a polynomial time algorithm for the group intersection problem, and vice versa. Group intersection is a natural generalization of the problem of determining the automorphism group of a graph.

On the lower level of difficulty is graph isomorphism. Because of [23], we choose here as representative *graph automorphism*, the problem of determining a generating set for the automorphism group of a graph. Note that for graphs, isomorphism and automorphism are problems of equal difficulty.

Although not an object of traditional mathematical inquiry, one may define *triple cosets* associated with three subgroups A, B, and C of a group G

$$A\pi B\psi C = \{\alpha\pi\beta\psi\gamma \mid \alpha \in A, \beta \in B, \gamma \in C\},$$

where $\pi, \psi \in G$. Testing membership in the triple coset ABC is an **NP**-complete problem [22]. Thus we have the following situation: Testing membership in a single coset (i.e. in $A\pi$) is in **P** [8]. Testing membership in a double coset is, as we conjecture, of intermediate difficulty. Testing membership in a triple coset is **NP**-complete.

There are a number of group-theoretic problems polynomial time equivalent to the double coset membership problem. One of these is the problem of determining the centralizer of a permutation group in another permutation group. Let A and B be two permutation groups. The *centralizer* of A in B is the subgroup of B defined by

$$\mathscr{C}_B(A) = \{\beta \in B \mid (\forall \alpha \in A)(\alpha\beta = \beta\alpha)\}.$$

That is, $\mathscr{C}_B(A)$ contains all permutations in B which commute with every permutation in A. If B is S_n, the symmetric group of degree n, then we speak of the *centralizer* of A, dropping the reference to the group B. The centralizer of A in A is called the *center* of A.

While determining the centralizer of a group A in a group B is of difficulty equal to double coset membership and is therefore at least as hard as graph isomorphism, determining the centralizer of A and the center of A are two important and natural subproblems which seem to be substantially easier. In Section 4, we show how to construct generators for the centralizer of a group G in $O(n^2 \cdot k)$ steps, where n is the degree of G and k is the number of generators given to specify G. Note that both G and $\mathscr{C}_{S_n}(G)$ could be of order exponential in n, while k may be assumed to be $O(n \cdot \log n)$, without loss of generality. We solve this problem by representing the centralizer of G as the automorphism group of a family of directed multigraphs, for which we give an efficient isomorphism test.

As a benefit of the graphical representation, we are able to recover results about the group-theoretic structure of centralizers. In particular, the centralizer of a permutation group of degree n is the direct product of s constituent groups W_i acting on the blocks of a certain partition of the permutation domain of G. Furthermore, each group W_i is isomorphic to the wreath product of a group G_i by a symmetric group. Here each group G_i is isomorphic to every one of its transitive constituents which are regular groups.

We then use the centralizer algorithm to construct the center of G in $O(n^6)$ additional steps, by intersecting G and $\mathscr{C}_{S_n}(G)$. Since $\mathscr{C}_{S_n}(G)$ normalizes G, we can do the intersection using our polynomial time algorithm from [15].

This paper is organized as follows: In Section 2, we review basic definitions and results from group theory and from graph theory to the extent needed here. We also briefly review relevant computational techniques recently discovered or analyzed (Subsection 2.7). In Section 3, we define a number of problems apparently harder than graph isomorphism and give polynomial time reductions among them. Since some of the problems arise outside of group theory, we briefly discuss how else they can be motivated and explain why they are structurally similar to isomorphism or automorphism problems for graphs. For the apparently harder problems, we also give proof of the polynomial time equivalence of the corresponding existence and counting problems. In Section 4, we show how to construct the centralizer of a permutation group G of degree n specified by a generating set, and show how to obtain the center of G.

2. Definitions, Terminology, and Background

We establish the basic terminology used throughout the paper, and review the basic definitions and results required. For the benefit of the reader unfamiliar with elementary group theory, we review the group-theoretic material more extensively in Subsections 2.1–2.5. Some graph-theoretic concepts are defined in 2.6, and previous work on the complexity of group-theoretic algorithms to the extent we require is summarized in 2.7.

2.1. *Permutations and Permutation Groups*

We consider 1–1 maps π of a fixed finite set X onto itself. Such a map is called a *permutation* of X. The *image* of a *point* $x \in X$ under π is denoted x^π. The *product* $\pi\psi$ of the permutations π and ψ of X is defined by $x^{(\pi\psi)} = (x^\pi)^\psi$, for all $x \in X$.

A *finite permutation group acting* on the finite set X is a nonempty set G of permutations of X closed under product formation, that is, for all $\pi, \psi \in G$, $\pi\psi \in G$. In the following, we always assume implicitly the finiteness of G and X.

Let G be a permutation group acting on X. The cardinality of G, denoted $|G|$, is the *order* of G, and the cardinality of X is the *degree* of G. The set of all permutations of X is the *symmetric group* of X, and is denoted $\mathrm{Sym}(X)$. The *trivial group* I consists of the identity permutation of X only. Usually, we shall choose for X the standard domain $\{1, 2,..., n\}$, and write S_n for $\mathrm{Sym}(X)$, in this case.

Any permutation can be written in *cycle notation*. A *cycle* of $\pi \in S_n$ is a list $(i_1, i_2,..., i_r)$ of distinct points in $\{1,..., n\}$, such that $i_1^\pi = i_2$, $i_2^\pi = i_3,..., i_{r-1}^\pi = i_r$, $i_r^\pi = i_1$. Any permutation can be written as the product of disjoint cycles. This product is unique up to a cyclic ordering within each cycle, and up to the ordering of the cycles themselves. When writing π, we usually omit the cycles of length 1. We denote the identity permutation with the empty cycle ( ) throughout.

2.2. *Subgroups, Right Cosets, Lagrange's Theorem*

Let G be a permutation group, H a nonempty subset of G, not necessarily a proper one. If H is also a permutation group, then H is a *subgroup* of G, written $H < G$. Note that $G < G$ and $I < G$.

Let $H < G$, $\pi \in G$. The set

$$H\pi = \{\chi\pi \mid \chi \in H\}$$

is a subset of G, called a *right coset* of H in G. Two right cosets $H\pi$ and $H\psi$ are either disjoint or equal, thus G can be *partitioned* into right cosets of H. This partitioning is written

$$G = H\pi_1 + H\pi_2 + \cdots + H\pi_r.$$

Note that H is a right coset of itself (in G), so we usually choose ( ) as the representative π_1 above. The *index* of H in G is the number of right cosets of H into which G is partitioned, and is written $(G : H)$. The cardinality of any right coset of H is equal to the order of H. We thus obtain

THEOREM 2.1 (Lagrange). *The order of G is equal to the product of the order of H and the index of H in G, i.e., $|G| = |H| \cdot (G : H)$.*

Let $H < G$ and $G = H\pi_1 + H\pi_2 + \cdots + H\pi_r$. Then the set $\{\pi_1, \pi_2, ..., \pi_r\}$ is a *complete right transversal* for H in G. In general, this set is not unique, but its cardinality is always equal to $(G : H)$.

2.3. *Orbits, Stabilizers, Generating Sets, Wreath Products*

Let $G < \mathrm{Sym}(X)$, and let $x \in X$ be a point in the permutation domain. The set

$$x^G = \{y \in X \mid y = x^\pi, \pi \in G\}$$

is called the *orbit* of x in G. The *length* of the orbit x^G is the cardinality $|x^G|$. The *stabilizer* of x in G is the subgroup G_x of G, where

$$G_x = \{\pi \in G \mid x^\pi = x\}.$$

There is an important correspondence between the points in the orbit of x in G and the right cosets of G_x in G [13, Theorem 5.2.2]:

THEOREM 2.2. *Let $G = G_x + G_x\pi_2 + \cdots + G_x\pi_r$. Then $|x^G| = r$, and, for all $\psi \in G_x\pi$, $x^\pi = x^\psi$.*

Note that $r \leqslant n$. Thus, the index of G_x in G is always *small.*

If Y is a subset of X, then the *pointwise stabilizer* of Y in X is the subgroup $G_{[Y]}$ of G, where

$$G_{[Y]} = \{\pi \in G \mid (\forall x \in Y)(x^\pi = x)\}.$$

Note that $G_{[X]} = I$, the trivial group.

Let $G < \mathrm{Sym}(X)$. A *generating set* for G is any subset K of G with the property that every element π of G can be expressed as a finite product of elements in K. In general, G has many generating sets.

Conversely, let K be any nonempty subset of $\mathrm{Sym}(X)$. Then the *group generated by* K is the smallest subgroup of $\mathrm{Sym}(X)$ containing K as a subset, and is denoted $\langle K \rangle$. K is a generating set for $\langle K \rangle$.

An important application of pointwise stabilizers is in the following tower of subgroups of G. Let $G < S_n$, and let $Y_i = \{1,..., i\}$. Let $G^{(1)} = G$, $G^{(j+1)} = G_{[Y_j]}$, $1 \leqslant j \leqslant n$. Then we have

$$I = G^{(n+1)} < G^{(n)} < \cdots < G^{(2)} < G^{(1)} = G.$$

Let U_i be a complete right transversal for $G^{(i+1)}$ in $G^{(i)}$, $1 \leqslant i \leqslant n$. Then every element π of G can be expressed *uniquely* as the product $\pi = \psi_n \psi_{n-1} \cdots \psi_1$, where $\psi_i \in U_i$. Furthermore, if $|U_i| = n_i$, then $|G| = \prod_{i=1}^{n} n_i$. (See [8, 28]).

In Subsection 2.7, we review an algorithm for determining the sets U_i in time polynomial in n, given an arbitrary polynomial-sized generating set for G.

Let $G < \mathrm{Sym}(X)$, $H < S_n$. The *wreath product*, $G \wr H$, of G by H is a permutation group whose elements are $(n+1)$-tuples $(\pi_1,..., \pi_n; \psi)$, where $\pi_i \in G$ and $\psi \in H$. The element $(\pi_1,..., \pi_n; \psi)$ acts on n copies of X as follows: First, for each i, permute the ith copy of X according to π_i. Then permute the n copies of X according to ψ. It may be helpful to visualize this action as an automorphism of a tree T of height 2. The root of T has n sons, labeled 1 through n. The sons of vertex i in T are the points of the ith copy of X. Now π_i permutes the sons of vertex i in T, whereas ψ permutes the sons of the root of T along with the subtrees rooted in them.

The wreath product $G \wr H$ has order $|G|^n \cdot |H|$. It contains as normal subgroup the n-fold direct product of G with itself, and this subgroup is the setwise stabilizer of every copy of X in $G \wr H$.

2.4. *Intersection, Centralizer, Center, Conjugation*

Let $G, H < S_n$ be permutation groups of degree n. The *intersection* $G \cap H$ of G and H is again a group. In particular, if the standard domain $\{1,..., n\}$ is partitioned into two sets X and $\bar{X}$, then the group $G_X = G \cap \mathrm{Sym}(X) \times \mathrm{Sym}(\bar{X})$ is the *setwise stabilizer* of X (equivalently, of $\bar{X}$) in G, and is the subgroup

$$G_X = \{\pi \in G \mid (\forall x \in X)(x^\pi \in X)\}.$$

Note that $G_{[X]}$ is a subgroup of G_X.

The union $G \cup H$ of G and H is usually not a group. The *group-theoretic union*

$\langle G, H\rangle$ of G and H is the smallest subgroup of S_n containing both G and H. Thus $\langle G, H\rangle$ is the group generated by $G \cup H$.

If π and ψ are permutations in S_n, then $\pi\psi$ is, in general, different from $\psi\pi$. The permutations π and ψ *commute*, if $\pi\psi = \psi\pi$.

Let $G < S_n$. The set of all permutations in S_n which commute with every element of G is a group, and is called the *centralizer* $\mathscr{C}_{S_n}(G)$ of G. The *centralizer in H of G*, $\mathscr{C}_H(G)$, is $\mathscr{C}_{S_n}(G) \cap H$. When $H = G$, then $\mathscr{C}_G(G)$ is called the *center* of G. The center of G consists of all elements of G which commute with every element of G. In Section 3, we discuss the problem of determining $\mathscr{C}_H(G)$. In Section 4, we give a polynomial time algorithm for computing the centralizer (in S_n) of an arbitrary permutation group G, and for computing the center of G.

If $\pi, \psi \in S_n$, then the permutation $\pi^{-1}\psi\pi$ is called the *conjugate* of ψ under π. $\pi^{-1}\psi\pi$ is usually abbreviated by ψ^{π}. Conjugation with a fixed permutation is a 1–1 map, i.e., $\psi^{\pi} = \chi^{\pi}$ iff $\psi = \chi$. If π and ψ commute, then $\psi^{\pi} = \psi$ and $\pi^{\psi} = \pi$.

Let $G < S_n$ be a group. Then $G^{\pi} = \{\psi^{\pi} \mid \psi \in G\}$ is again a group and is isomorphic to G, since, for $\psi, \chi \in G$, $\psi^{\pi}\chi^{\pi} = (\psi\chi)^{\pi}$. In Subsection 2.7, we review an algorithm for computing G^{π} from G and from π.

Let $G, H < S_n$ be two groups: G *normalizes* H if, for all $\pi \in G$, $H^{\pi} = H$. Equivalently, G normalizes H iff, for all $\pi \in G$ and $\chi \in H$, $\chi^{\pi} \in H$.

2.5. *Double Cosets*

Let A and B be subgroups of the permutation group G. For $\pi \in G$, define the subset of G

$$A\pi B = \{\alpha\pi\beta \mid \alpha \in A, \beta \in B\}.$$

Then $A\pi B$ is a *double coset* of A and B in G. Two double cosets $A\pi B$ and $A\psi B$ are either disjoint or equal. The cardinality of $A\pi B$ is given by

THEOREM 2.3 ([13, Theorem 1.7.1]). $|A\pi B| = |A| \cdot |B|/|A^{\pi} \cap B|$.

In particular, the double coset AB, i.e., the double coset $A\pi B$ where $\pi = (\ )$, has the cardinality $|AB| = |A| \cdot |B|/|A \cap B|$.

A group G may be partitioned into double cosets of A and B. Define an equivalence relation on G by $\pi \equiv \psi$ if there exists $\alpha \in A$ and $\beta \in B$, such that $\pi = \alpha\psi\beta$. The double cosets of A and B are then the equivalence classes in G.

Note that double cosets need not be of uniform cardinality, and so there is no equivalent of Lagrange's Theorem for double cosets. Computationally, double cosets appear to be difficult objects. In Section 3, we discuss the basic computational questions associated with double cosets and interpret double cosets as isomorphism classes of certain structures.

2.6. *Graphs*

We consider directed graphs. A *directed graph* is a pair $X = (V, E)$, where V is a finite set of *vertices*, and E is a subset of $V \times V$. If $(v, w) \in E$, then (v, w) is a *directed edge* from vertex v to vertex w. We only consider graphs without selfloops, i.e., there are no edges (v, v). In the following, *graph* will always mean *directed graph*, unless specifically stated otherwise.

A *directed multigraph* is a pair $X = (V, E)$, where V is a finite set of vertices and E is a finite multiset of ordered pairs (v, w), where v and w are distinct vertices in V. A multigraph differs from a graph in that multiple edges from v to w are possible. In Section 4, we show that the centralizer of any permutation group G arises as the automorphism group of an edge-colored multigraph.

Let $X = (V, E)$ be a graph, $v \in V$ a vertex of X. The *indegree* of v is the number of edges (u, v) in E, and the *outdegree* of v is the number of edges (v, w) in E. Intuitively, the indegree is the number of edges ending in a vertex, the *outdegree* the number of edges originating in a vertex.

Let $X = (V, E)$ be a graph, v and w vertices in X, not necessarily distinct. A *path between v and w* in X is a sequence of graph vertices $u_1, u_2, \ldots, u_k$, $1 \leqslant k$, such that $v = u_1$, $w = u_k$, and X has edges (u_i, u_{i+1}) or (u_{i+1}, u_i), $1 \leqslant i < k$. If, furthermore, there are edges (u_i, u_{i+1}) for all $i < k$, then $u_1, u_2, \ldots, u_k$ is a *directed path from v to w* in X. If $k = 1$, then the path is *trivial*. If the u_i are all distinct vertices of X, then the path is *simple*. A *cycle* is a directed path $u_1, \ldots, u_k$, such that $u_1 = u_k$ and $u_1, \ldots, u_{k-1}$ is a simple path.

Let $X = (V, E)$ be a graph, V' a subset of V and E' a subset of E. If $Y = (V', E')$ is a graph, then Y is called a *subgraph* of X. A graph X is a *connected* graph if, for every two vertices v and w of X, there is a path in X between v and w. Further, X is *strongly connected* if, for any two vertices v and w of X, there is a directed path in X from v to w and a directed path from w to v.

Let $X = (V, E)$ be a graph, $Y = (V', E')$ a subgraph of X. Then Y is a *component* of X, if Y is a connected graph and, for every vertex v in Y and w in X but not in Y, there is no path in X between v and w. The components of any graph can be found in $O(|V| + |E|)$ steps.

We shall consider the problem of testing certain graphs for isomorphism, and the problem of determining their automorphism group.

Let $X = (V, E)$ and $X' = (V', E')$ be two graphs. Then X and X' are *isomorphic* if there is a bijective map ι from V onto V' such that (v, w) is in E whenever (v^ι, w^ι) is in E'. Note that v^ι denotes the image of v under ι. A bijective map π from V onto V (i.e., a permutation π of V) is an *automorphism* of X if (v, w) is an edge of X whenever (v^π, w^π) is also an edge of X. The set of all automorphisms of a graph X is a permutation group acting on the vertex set of X, and is denoted $\mathrm{Aut}(X)$. The group $\mathrm{Aut}(X)$ is called the *automorphism group* of X. If $X' = (V, E')$ is a graph isomorphic to $X = (V, E)$, where both graphs have the same vertex set, then the set of all distinct isomorphisms from X to X' is a right coset of $\mathrm{Aut}(X)$ [23].

2.7. *Computational Techniques from Previous Related Work*

When investigating the complexity of algorithms for permutation groups, one of the most fundamental issues which needs to be addressed is how to represent groups. Since the order of a permutation group G of degree n may be as large as $n!$, it is obviously not advantageous to represent G by a list of its elements, except when G is known to be of small order. It turns out, however, that G always has a succinct generating set (Theorem 2.4 below). So, we may wish to learn which computations in permutation groups can be carried out efficiently, based on representing groups by succinct generating sets. In this section, we summarize previous results pertaining to that question, to the extent required for our paper.

THEOREM 2.4. *If G is a permutation group of order m, then G is generated by a subset K of G of size at most* $\lceil \log m \rceil$.

The theorem is easy to prove and is well known. As a consequence of Theorem 2.4, any permutation group of degree n can be generated from $\mathrm{O}(n \cdot \log n)$ permutations.

Previous work on the complexity of determining groups from generating sets of permutations [27, 28] has resulted in

THEOREM 2.5 (Sims, Furst, Hopcroft, Luks). *Let G be a permutation group of degree n. Then there is a generating set K_0 of G of size at most* $\mathrm{O}(n^2)$ *with the following properties*:

(1) *Every element of G can be expressed uniquely as product of exactly n elements of K_0.*

(2) *From K_0, membership in G can be tested in* $\mathrm{O}(n^2)$ *steps.*

(3) *The order of G can be determined in* $\mathrm{O}(n^2)$ *steps.*

The special generating set K_0 of Theorem 2.5 may be chosen to consist precisely of complete right transversals for the groups $G^{(i+1)}$ in $G^{(i)}$, $i \leqslant n$, in the tower

$$I = G^{(n+1)} < G^{(n)} < \cdots < G^{(1)} = G,$$

where $G^{(i+1)}$ is the pointwise stabilizer of $\{1,..., i\}$ in G. The set K_0 can be found using

THEOREM 2.6 (Sims, Furst, Hopcroft, Luks). *Let G be a permutation group of degree n given by the generating set K. Then the set K_0 of Theorem* 2.5, *generating G as well, can be found in* $\mathrm{O}(n^2 \cdot |K| + n^6)$ *steps.*

The method was originally discovered by Sims, e.g., [28]. The analysis is due to Furst *et al.* [8], who rediscovered the method independently. We interpret Theorems 2.5 and 2.6 as saying that the representation of a permutation group by a small generating set is useful for determining, computationally, the most basic properties of the group.

Given generating sets for groups $G, H < S_n$, it is an open problem whether generators for $G \cap H$ can be found in polynomial time (Problem 3.5). But there are a number of special situations for which there exist efficient intersection algorithms [8, 15, 21]. We shall need here our algorithm for intersecting with a normalizing group [15].

THEOREM 2.7. *Let $G, H < S_n$ be permutation groups of degree n presented by generating sets of size* $O(n^2)$. *Then*

(1) *In* $O(n^6)$ *steps, we can test whether G normalizes H.*

(2) *If G normalizes H, then generators for $G \cap H$ can be found in* $O(n^6)$ *steps.*

We shall use this result in Section 4. In conclusion, we make the following simple observations:

PROPOSITION 2.1. *If $G = \langle K_G \rangle$, $H = \langle K_H \rangle$, then $\langle G, H \rangle = \langle K_G \cup K_H \rangle$.*

PROPOSITION 2.2. *If $G < S_n$, $\pi \in S_n$, and $G = \langle K \rangle$, then $G^\pi = \langle K^\pi \rangle$.*

As a consequence, we can determine the group-theoretic union and conjugate groups in polynomial time, using the techniques of Theorems 2.5 and 2.6.

3. GROUP-THEORETIC PROBLEMS GENERALIZING GRAPH ISOMORPHISM

We examine group-theoretic problems which are at least as hard as graph isomorphism, since graph isomorphism can be polynomial time reduced to every one of them. All problems to be discussed are clearly in **NP**. Furthermore, the existence problems are polynomial time equivalent to the associated counting problems. For **NP**-complete existence problems, the associated counting problems are believed to be more difficult [12, 29]. Therefore, we do not believe that these problems are **NP**-complete.

We divide our exposition into subsections. In the first subsection, we state the double coset membership problem, and explain why this problem generalizes graph isomorphism. We prove that for this problem, counting and existence are polynomial time equivalent. In the second subsection, we state the group intersection problem and several problems polynomially equivalent to it. Double coset membership and group intersection are shown to be polynomial time equivalent.

In Subsection 3.3, we give a group-theoretic formulation of graph automorphism. While this problem is clearly reducible to the preceding problems, we have not been able to find a polynomial time reduction in the converse direction, and we suspect that the problems in Subsections 3.1 and 3.2 are harder. We further remark on this conjecture in Subsection 3.4.

Throughout this section, we assume that groups are presented by generating sets of size polynomial in the degree of the groups. Because of Theorems 2.4–2.6, this constitutes no loss of generality.

3.1. *Double Coset Problems*

Recall from Subsection 2.5 that double cosets may be understood as the equivalence classes of an equivalence relation induced on S_n by the subgroups A and B of S_n. That is, $\pi, \psi \in S_n$ are equivalent iff there exist $\alpha \in A$ and $\beta \in B$ such that

$\psi = \alpha\pi\beta$. A natural question to ask is the complexity of testing equivalence of ψ and π, given generating sets for A and for B. We call this test the *Double Coset Membership Problem.* The problem is a generalized isomorphism question with instances arising in combinatorial counting problems, and we explain this interpretation below. Double coset membership is clearly in **NP**.

As a subproblem, we consider testing whether π is equivalent to the identity permutation ( ), i.e., whether $\pi = \alpha\beta$ for some $\alpha \in A$, $\beta \in B$. We will show that this *Group Factorization Problem* is of difficulty equal to double coset membership.

The counting problem associated with group factorization is to determine the number of distinct factorizations $\alpha_i\beta_i$ of π over A and B. We show that this problem is no harder than group factorization.

There are no known **NP**-complete existence problems whose associated counting problems are also **NP**-complete. Furthermore, the counting problems associated with many **NP**-complete existence problems have been shown to be #**P**-complete. We therefore view the polynomial time equivalence of group factorization and its associated counting problem as evidence against the possibility that the double coset membership problem is **NP**-complete.

PROBLEM 3.1 (Double Coset Membership). Given the groups $A, B < S_n$ by generating sets and the permutations $\pi, \psi \in S_n$, test whether $\psi \in A\pi B$.

PROBLEM 3.2 (Group Factorization). Given the groups $A, B < S_n$ by generating sets and the permutation $\pi \in S_n$, test whether there are $\alpha \in A$, $\beta \in B$, such that $\pi = \alpha\beta$. Equivalently, test whether $\pi \in AB$.

PROBLEM 3.3 (Number of Factorizations). Given the groups $A, B < S_n$ by generating sets and the permutation $\pi \in S_n$, determine the number $k \geqslant 0$ of distinct factorizations $\pi = \alpha\beta$ of π, where $\alpha \in A$, $\beta \in B$.

For completeness sake, we formally state the obvious.

PROPOSITION 3.1. *Problem* 3.1 *is in* **NP**.

Problem 3.1 has the following instance familiar from combinatorics. We are given a combinatorial structure X with n points, where X has the known automorphism group B. For example, X could be a graph with n vertices and with the automorphism group B. Furthermore, we are given a partition of n into k positive numbers n_i, i.e., $n = n_1 + n_2 + \cdots + n_k$. We wish to color the points of X with k distinct colors c_i forming the color set $C = \{c_1, ..., c_k\}$.

Let us call a coloring (X, λ) of the points of X *admissible* if λ is a map from the points of X into C such that exactly n_i points of X are mapped into c_i. We consider the question whether two admissible colorings (X, λ) and (X, μ) are *equivalent* in the sense that there is a symmetry $\beta \in B$ of X such that β maps the colored structure (X, λ) into (X, μ). That is, for all points z in X, the color $\lambda(z)$ should be $\mu(z^\beta)$.

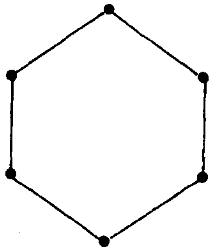

FIGURE 3.1

EXAMPLE 3.1. Let X be the graph shown below in Fig. 3.1; it has 6 vertices. We consider the partition $3 + 3$ of 6. The two admissible colorings shown in Fig. 3.2 are equivalent, whereas the two admissible colorings in Fig. 3.3 are not. ▮

To see the relationship between equivalent admissible colorings and double coset membership, we let permutations $\pi \in S_n$ specify admissible colorings as follows: The permutation $\pi \in S_n$ will specify the labelling (X, λ) by prescribing that the points $1^\pi, 2^\pi, \ldots, n_1^\pi$ of X are mapped into the color c_1, points $(n_1 + 1)^\pi, \ldots, (n_1 + n_2)^\pi$ into c_2, etc. It is clear that this labelling is admissible. Equivalently, we may think of the labelling specified by π as arising by *superimposing* a *label structure* Y with n points on the structure X, where Y consists just of n distinct points with points $\{1, 2, \ldots, n_1\}$

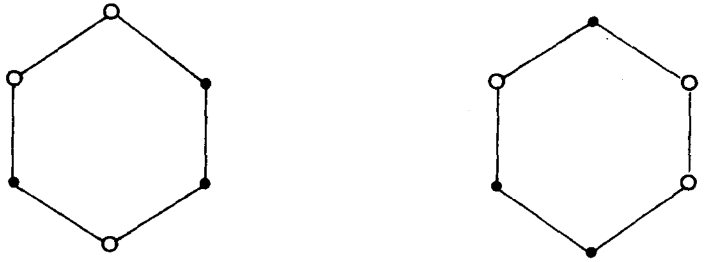

FIGURE 3.2

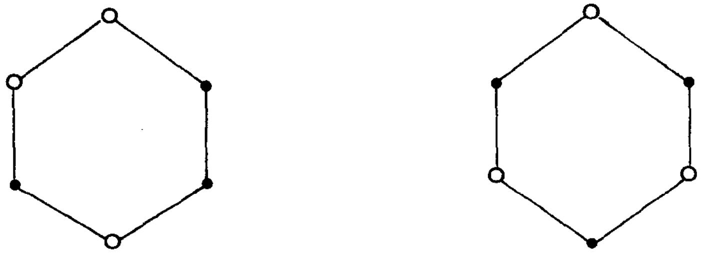

FIGURE 3.3

being of color c_1, points $\{n_1+1,\dots, n_1+n_2\}$ of color c_2, etc., and points $\{n-n_k+1,\dots, n\}$ of Y being of color c_k. Now π specifies how to superimpose Y on X.

Recall that $B < S_n$ is the symmetry group of X, and let $A < S_n$ be the symmetry group of Y. In our example, A is the direct product of symmetric groups of degree $n_1, n_2,\dots, n_k$, respectively. For $\alpha \in A$, it is clear that $\alpha\pi$ and π specify the same labelling of X, thus π and $\alpha\pi$ are equivalent labellings. Furthermore, for $\beta \in B$, π and $\pi\beta$ must specify equivalent labellings. Now it is not hard to see that the double coset $A\pi B$ specifies all admissible labellings of X equivalent to π.

Therefore, in general, $A\pi B$ contains all *isomorphic* ways of superimposing a structure Y with automorphism group A on a structure X with automorphism group B. We amplify on this interpretation by explaining how graph isomorphism may be so formulated.

Let $X=(V,F)$ and $X'=(V,F')$ be two (undirected) graphs with the vertex set $V=\{1,\dots, n\}$ and an equal number of edges. Consider the complete graph $K_n=(V,E_n)$ and note that both X and X' arise from K_n when labelling each edge in E_n with one of the two colors: *edge* and *not an edge.*

Let B be the permutation group induced in $\mathrm{Sym}(E_n)$ by the action of S_n on the edges of the complete graph K_n. Consider the partition of $E_n=F+\bar{F}$, and let π be any permutation in the group $A=\mathrm{Sym}(F)\times\mathrm{Sym}(\bar{F})$. Note that π specifies the graph X in the sense of superimposing the partition $F+\bar{F}$ of E_n as a labeling structure on the edges of K_n. Next, let ψ be any permutation in $\mathrm{Sym}(E_n)$ which maps F onto F' and $\bar{F}$ onto $\bar{F}'$, where $\bar{F}'=E_n-F'$. Note that ψ specifies the graph X'. Moreover, the groups A and B, as well as the permutations π and ψ, are easily obtained from the graphs X and X'. Then $\psi\in A\pi B$ iff X and X' are isomorphic graphs. That is, graph isomorphism is a special case of the double coset membership problem. Thus, we may consider double cosets as abstract isomorphism classes.

We now show that Problems 3.1–3.3 are of equal difficulty, i.e., polynomial time equivalent.

THEOREM 3.1. *Problems* 3.1 *and* 3.2 *are polynomial time equivalent.*

Proof. Since Problem 3.2 is a special case of Problem 3.1, we only need to reduce Problem 3.1 to Problem 3.2. For this reduction, we observe first that the elements $\alpha\pi\beta$ of $A\pi B$ may be put into 1–1 correspondence with the elements $\pi^{-1}\alpha\pi\beta$ of $A^\pi B$. Thus, $\psi\in A\pi B$ iff $\pi^{-1}\psi\in A^\pi B$. By Theorems 2.5, 2.6, and Proposition 2.2, this establishes a polynomial time reduction. ∎

In order to show the polynomial time equivalence of Problems 3.2 and 3.3, we need

LEMMA 3.1. *If* $\pi=\alpha_1\beta_1=\alpha_2\beta_2=\cdots=\alpha_k\beta_k$, $\alpha_i\in A$, $\beta_i\in B$, *are the distinct factorizations of* π *over* A *and* B, *then* $k=|A\cap B|$.

Proof. Let $C=A\cap B$. Since $\alpha_i\beta_i=\alpha_j\beta_j$, we have $\alpha_j^{-1}\alpha_i=\beta_j\beta_i^{-1}$, and so β_i and β_j

are in the same right coset of C. Furthermore, if $\pi = \alpha\beta$ and $\gamma \in C$, then also $\pi = \alpha\gamma^{-1}\gamma\beta = \alpha'\beta'$, thus the β_i form a right coset of C. Finally, observe that α_i is uniquely determined by π and β_i, thus $k = |C|$. ▮

THEOREM 3.2. *Problems* 3.2 *and* 3.3 *are polynomial time equivalent.*

Proof. It is clear that we can reduce Problem 3.2 to Problem 3.3, in polynomial time.

We establish the opposite reduction by a two-step algorithm. First, test whether $\pi \in AB$ using the algorithm for Problem 3.2. This determines whether to output zero or a positive number. Second, if $\pi \in AB$, determine $|C|$ with the algorithm below, which makes repeated calls on the algorithm for Problem 3.2.

Let $C = A \cap B$, and let $A^{(i)}$, $B^{(i)}$, and $C^{(i)}$ be the pointwise stabilizers of $\{1,..., i-1\}$ in A, B, and C, respectively. Here $A^{(1)} = A$, $B^{(1)} = B$, $C^{(1)} = C$, and $A^{(n+1)} = B^{(n+1)} = C^{(n+1)} = I$. By Theorem 2.6, we may assume that we have constructed complete right transversals U_i and V_i for $A^{(i+1)}$ in $A^{(i)}$ and for $B^{(i+1)}$ in $B^{(i)}$, respectively, in polynomial time. We shall determine the orbit Δ_i of i in $C^{(i)}$. Note that $|\Delta_i| \leqslant n-i+1$. Having done so, we can determine $|C|$ from the formula $|C| = \prod_{i=1}^{n} |\Delta_i|$, in polynomial time.

Let $\pi_i \in U_i$ be a right coset representative for $A^{(i+1)}$ in $A^{(i)}$ mapping i into j, i.e., with $i^{\pi_i} = j$. We do the following:

(a) Find $\psi_i \in V_i$ such that $i^{\psi_i} = j$. If there is no such ψ_i, then j cannot be in Δ_i.

(b) Let $\chi = \pi_i \psi_i^{-1}$, where ψ_i was found in Step (a). Then $j \in \Delta_i$ iff $\chi \in A^{(i+1)}B^{(i+1)}$.

The correctness of Step (a) follows trivially from $C^{(i)} = A^{(i)} \cap B^{(i)}$ and Theorem 2.2. For Step (b), observe that $j \in \Delta_i$ iff there are representatives $\pi_k \in U_k$ and $\psi_k \in V_k$, $i \leqslant k \leqslant n$, such that

$$\varphi = \pi_n \pi_{n-1} \cdots \pi_i = \psi_n \psi_{n-1} \cdots \psi_i \in C^{(i)}.$$

Thus, $j \in \Delta_i$ iff

$$\chi = \pi_i \psi_i^{-1} = \pi_{i+1}^{-1} \cdots \pi_n^{-1} \psi_n \psi_{n-1} \cdots \psi_{i+1} = \pi' \psi' \in A^{(i+1)}B^{(i+1)}.$$

For the timing of this reduction, observe that we test at most $n-i+1$ products $\pi_i \psi_i^{-1}$ for membership in $A^{(i+1)}B^{(i+1)}$, thus the reduction is polynomial time. ▮

The significance of Theorem 3.2 is that it can be interpreted as technical evidence against the possibility that Problems 3.1 and 3.2 are **NP**-complete (cf. [12, 29]), as discussed above.

A problem not considered here but related to Problem 3.1 is the question of determining the number of double cosets $A\pi B$ into which S_n is partitioned. This is the combinatorial question of how many *nonisomorphic* ways exist of superimposing the labelling structure Y with symmetries A on the structure X with symmetries B. A special case of this problem is the question of how many *nonisomorphic* graphs with n vertices and p edges exist. At present, we do not know if this question is even in **NP**. An obviously exponential algorithm for the problem is contained in [5].

3.2. *Intersection Problems*

There is a 1–1 correspondence between the right cosets of A contained in the double coset AB and the right cosets of $A \cap B$ in B, which gives rise to Theorem 2.3 and to Lemma 3.1. This relatedness of AB and $A \cap B$ is also reflected in the interpretation of double cosets and of group intersection, for group intersection may be interpreted as constructing the automorphism group of the structure resulting from superimposing a structure Y with automorphism group A on a structure X with automorphism group B.

We have exploited the relationship between AB and $A \cap B$ in the reduction given in the proof of Theorem 3.2, when determining $|A \cap B|$. A little additional effort can produce a polynomial time reduction of the problem of determining *generators* for $A \cap B$ to Problem 3.2, in Theorem 3.3 below. The reduction in the opposite direction uses as intermediate steps the Coset Intersection Triviality and the Setwise Stabilizer Problems. Note that there are other group-theoretic problems which seem to lie on the same level of difficulty [22].

PROBLEM 3.4 (Coset Intersection Triviality). Given the groups $A, B < S_n$ by generating sets, and given a permutation $\pi \in S_n$, test whether $A\pi \cap B$ is empty.

PROBLEM 3.5 (Group Intersection). Given the groups $A, B < S_n$ by generating sets, determine a generating set for $C = A \cap B$.

PROBLEM 3.6 (Setwise Stabilizer). Given the group $A < S_n$ by a generating set, and given a subset X of $\{1, ..., n\}$, determine a generating set for A_X, the setwise stabilizer of X in A.

PROBLEM 3.7 (Centralizer in Another Group). Given the groups $A, B < S_n$ by generating sets, determine a generating set for $\mathscr{C}_A(B)$, the centralizer of B in A.

THEOREM 3.3. *Problem* 3.5 *can be reduced to Problem* 3.1 *in polynomial time.*

Proof. We shall extend the algorithm of the proof of Theorem 3.2 and determine generators for $C = A \cap B$. As before, let $A^{(i)}$, $B^{(i)}$, and $C^{(i)}$ denote the pointwise stabilizers of $\{1, ..., i-1\}$ in A, B, and C, respectively, and recall that $A^{(1)} = A$, $B^{(1)} = B$, $C^{(1)} = C$, $A^{(n+1)} = B^{(n+1)} = C^{(n+1)} = I$. Let U_i and V_i be complete right transversals for $A^{(i+1)}$ in $A^{(i)}$ and for $B^{(i+1)}$ in $B^{(i)}$. We will determine complete right transversals W_i for $C^{(i+1)}$ in $C^{(i)}$, where $1 \leqslant i \leqslant n$. The algorithm to be described has much similarity with Sims' backtracking algorithm for intersecting permutation groups, [15, 28]. However, by using an algorithm for Problem 3.1 as a subroutine, we can eliminate the backtracking.

For each $\pi_i \in U_i$, we determine first whether $j = i^{\pi_i}$ is a point in Δ_i, the orbit of i in $C^{(i)}$. We use here Steps (a) and (b) from the proof of Theorem 3.2. We outline how to find a coset representative $\varphi_i \in W_i$ mapping i into j whenever Step (b) determines that j is in Δ_i. Let π_i and ψ_i be the representatives found in Step (b), and recall that

$$\chi = \pi_i \psi_i^{-1} \in A^{(i+1)}B^{(i+1)} \qquad \text{iff} \qquad \pi_i \psi_i^{-1} = \pi_{i+1}^{-1}\pi_{i+2}^{-1} \cdots \pi_n^{-1}\psi_n \cdots \psi_{i+2}\psi_{i+1},$$

where $\pi_{i+k} \in U_{i+k}$ and $\psi_{i+k} \in V_{i+k}$. Therefore,

$$\chi_i = \pi_i \psi_i^{-1} \in A^{(i+1)}B^{(i+1)}$$

iff

$$\chi_{i+k} = \pi_{i+k}\pi_{i+k-1} \cdots \pi_i \psi_i^{-1} \cdots \psi_{i+k-1}^{-1}\psi_{i+k}^{-1} \in A^{(i+k+1)}B^{(i+k+1)}.$$

We therefore proceed as follows: Let $\chi_i = \pi_i \psi_i^{-1} \in A^{(i+1)}B^{(i+1)}$ be the product of the pair of coset representatives determined by Step (b), and set k to i.

(c) Find a pair $\pi_{k+1} \in U_{k+1}$, $\psi_{k+1} \in V_{k+1}$, such that $\pi_{k+1}\chi_k\psi_{k+1}^{-1} \in A^{(k+2)}B^{(k+2)}$. Note that such a pair must always exist and that it can be found by making at most $n-k$ calls to the algorithm for Problem 3.1.

Repeat Step (c), letting $\chi_{k+1} = \pi_{k+1}\chi_k\psi_{k+1}^{-1}$ for the found pair, and increase k until we have determined $\chi_n = (\ )$. The desired coset representative is now $\pi_n \pi_{n-1} \cdots \pi_i$, where the π_k have been found in Step (c) above.

This procedure makes at most $O(n^3)$ calls on the algorithm for Problem 3.1. Therefore, we can determine the sets W_i forming a generating set for C by a polynomial time reduction to Problem 3.1. ▮

For the reduction establishing that Problem 3.2 can be polynomial time reduced to Problem 3.5, we proceed in the following steps: First, we show that Problems 3.2 and 3.4 are polynomially equivalent. Next, we observe that Problem 3.6 is a special case of Problem 3.5, and finally, we show how to reduce Problem 3.4 to Problem 3.6.

PROPOSITION 3.2. *Problems* 3.2 *and* 3.4 *are polynomially equivalent.*

Proof. If $\pi \in AB$, then $\pi = \alpha\beta$, hence $\alpha^{-1}\pi \in B$, and so $A\pi \cap B$ is not empty. Conversely, if $\varphi \in A\pi \cap B$, then $\varphi = \alpha\pi = \beta$, for some $\alpha \in A$ and some $\beta \in B$, hence $\pi \in AB$. ▮

PROPOSITION 3.3. *Problem* 3.6 *can be polynomial time reduced to Problem* 3.5.

Proof. Observe that for $A < S_n$, X a proper subset of $\{1,\ldots, n\}$ and $\bar{X} = \{1,\ldots, n\} - X$, $A_X = A \cap \operatorname{Sym}(X) \times \operatorname{Sym}(\bar{X})$. ▮

THEOREM 3.4 (Lipton, Kannan). *Problem* 3.4 *can be reduced to Problem* 3.6 *in polynomial time.*

Proof (Luks). We wish to test whether $A\pi \cap B$ is empty, given an algorithm for

Problem 3.6. We assume that $A, B < \mathrm{Sym}(X)$, and consider the group $D = \{(\alpha, \beta) \mid \alpha \in A, \beta \in B\}$ acting on $X \times X$ by the rule $(x, y)^{(\alpha, \beta)} = (x^\alpha, y^\beta)$. Consider the sets $Z = \{(x, x) \mid x \in X\}$ and $Z' = \{(x^{\pi^{-1}}, x) \mid x \in X\}$. We test in the manner described below whether D contains an element δ such that $Z^\delta = Z'$ using the algorithm for Problem 3.6. Clearly this is the case iff $A\pi \cap B$ is not empty.

So, consider the problem of finding an element δ in the group $D < \mathrm{Sym}(Y)$ which maps the subset Z of Y onto the subset Z' of Y. Here we construct the group G as the wreath product, $D \wr C_2$, of D by the cyclic group of order 2. Recall that G has $D \times D$ as subgroup of index 2, the setwise stabilizer of Y_1 in G.

Using an algorithm for Problem 3.6, we determine a generating set for the subgroup G' of G which stabilizes setwise $Z_1 \cup Z_2'$, where Z_1 is the subset Z in the copy Y_1, Z_2' the subset Z' in the copy Y_2 of Y. Let $H = G' \cap D \times D$ be the setwise stabilizer of Y_1 in G'. Since $(G: D \times D) = 2$, the index of H in G' is at most 2. Therefore, using the techniques of [8], we can determine H from a generating set for G'. We conclude the proof of the theorem by showing that $(G' : H) = 2$ iff D contains an element δ which maps Z onto Z'.

Assume there exists an element $\delta \in D$ such that $Z^\delta = Z'$. Then $\psi = (\delta, \delta^{-1}; (1, 2))$ must exchange Z_1 with Z_2', hence $\psi \in G'$. Since ψ does not stabilize Y_1, the index of H in G' must be 2. Conversely, let $(G' : H) = 2$ and consider an element $\psi \in G'$ which is not in H. Then ψ must be of the form $(\delta, \gamma; (1, 2))$. Since $Z_1 \subset Y_1$ and $Z_2' \subset Y_2$, Z_1^δ must be Z_1', i.e., δ is the desired element of D. ▮

It is interesting to note the similarity in character between the proof of Theorem 3.4 and the proof that graph isomorphism is polynomially reducible to graph automorphism. For graphs Z and Z', one forms the disjoint union $Z + Z'$ of the two graphs and determines separately the orders of the automorphism groups, $|\mathrm{Aut}(Z)|$, $|\mathrm{Aut}(Z')|$, and $|\mathrm{Aut}(Z + Z')|$. Then it is easy to see that Z and Z' are isomorphic iff $|\mathrm{Aut}(Z + Z')| = 2 \cdot |\mathrm{Aut}(Z)| \cdot |\mathrm{Aut}(Z')|$. In case the graphs are isomorphic, $\mathrm{Aut}(Z + Z')$ is isomorphic to the wreath product of $\mathrm{Aut}(Z)$ by C_2. Hence the reduction determines whether the subgroup $\mathrm{Aut}(Z) \times \mathrm{Aut}(Z')$ has index 2 in $\mathrm{Aut}(Z + Z')$.

We finally establish the polynomial time equivalence of Problem 3.7 with Problems 3.4–3.6. Our first step will be to give a polynomial time reduction of Problem 3.7 to Problem 3.5, group intersection. For this, we need to understand the structure of the centralizer (in S_n) of a cyclic group Z generated by a permutation $\pi \in S_n$. The structure of $\mathscr{C}_{S_n}(Z)$ is derived using elementary combinatorial arguments which seem somewhat tedious in detail. For this reason, we give examples following each of the key lemmas to illustrate the constructions.

Lemma 3.2. *Let $\pi, \psi \in S_n$ be permutations, where π, in cycle notation, is $\pi = (i_1, \ldots, i_s)(i_{s+1}, \ldots, i_t) \cdots (i_q, \ldots, i_r)$. Then the conjugate of π under ψ is the permutation $\pi^\psi = (i_1^\psi, \ldots, i_s^\psi)(i_{s+1}^\psi, \ldots, i_t^\psi) \cdots (i_q^\psi, \ldots, i_r^\psi)$.*

Proof. [13, Lemma 5.1.1].

EXAMPLE 3.2. Let $\pi = (1,2)(3,4)(6,7,9)$ and $\psi = (1,2,3,4)(5,6)$ be permutations in S_9. Then $\pi^\psi = (1^\psi, 2^\psi)(3^\psi, 4^\psi)(6^\psi, 7^\psi, 9^\psi) = (2,3)(4,1)(5,7,9)$, which is readily verified. ▮

LEMMA 3.3. *Let $A < S_n$ be generated by $\{a_1, a_2, \ldots, a_p\}$, and let C_i be the set of all permutations in S_n which commute with a_i, i.e., $C_i = \{\pi \in S_n \mid \pi^{-1} a_i \pi = a_i\}$. Then $\mathscr{C}_{S_n}(A) = \bigcap_{i=1}^p C_i$.*

Proof. It is easy to see that the sets C_i are groups. Therefore, $D = \bigcap_{i=1}^p C_i$ is also a group. Let $\gamma \in D$, $a \in A$. Since a is a finite product of the a_i, clearly $\gamma^{-1} a \gamma = a$, thus $\gamma \in \mathscr{C}_{S_n}(A)$. Conversely, let $\gamma \in \mathscr{C}_{S_n}(A)$. Since $a_i \in A$, $\gamma^{-1} a_i \gamma = a_i$, $1 \leqslant i \leqslant p$, thus $\gamma \in D$. Therefore, $\mathscr{C}_{S_n}(A) = D$. ▮

As a consequence of the lemma, $C_i = \mathscr{C}_{S_n}(a_i) = \mathscr{C}_{S_n}(Z_i)$, where Z_i is the cyclic group generated by a_i.

LEMMA 3.4. *Let $\pi \in S_n$ be a permutation, $Z = \langle \pi \rangle$ the cyclic group generated by π. Then generators for the centralizer $\mathscr{C}_{S_n}(Z)$ can be determined in polynomial time.*

Proof. Let π be written in cycle notation with the distinct cycle lengths $l_1, l_2, \ldots, l_r$, and with m_i cycles of length l_i. Here we also consider cycles of length 1. Then $\psi \in S_n$ commutes with π iff $\pi^\psi = \pi^{\psi^{-1}} = \pi$. By Lemmas 3.2 and 3.3, we therefore look for all those permutations ψ such that conjugation under ψ *rotates* the cycles of π and/or exchanges cycles of equal length. Since the cycle notation is unique up to those two rewriting rules, $\mathscr{C}_{S_n}(Z)$ must consist precisely of those permutations.

For each $i \leqslant r$, we examine the cycles of length l_i. By inspection, we shall produce a set K_i of permutations generating all those permutations ψ which commute with π and are such that all points which lie in cycles of π of length other than l_i are fixed by ψ.

Let $J_1, \ldots, J_{m_i}$ be all the cycles in π of length l_i. The set K_i will consist of permutations $\zeta_{i,1}, \ldots, \zeta_{i,m_i}$ and two permutations α_i and β_i; the set K_i contains no other permutation. We let $\zeta_{i,s} = J_s$, $1 \leqslant s \leqslant m_i$. Furthermore, if $J_s = (j_{s,1}, j_{s,2}, \ldots, j_{s,l_i})$, then $\alpha_i = (j_{1,1}, j_{2,1}) \cdots (j_{1,l_i}, j_{2,l_i})$, and $\beta_i = (j_{1,1}, j_{2,1}, \ldots, j_{m_i,1})(j_{1,2}, \ldots, j_{m_i,2}) \cdots (j_{1,l_i}, \ldots, j_{m_i,l_i})$.

Note that S_n is generated by $\alpha = (1,2)$ and $\beta = (1, 2, \ldots, n)$. Thus, it is clear that α_i and β_i generate permutations ψ which conjugate π by permuting the order of the cycles of length l_i in all possible ways. Since the $\zeta_{i,k}$ generate all rotations of these cycles, K_i generates all permutations $\psi \in S_n$ such that conjugation under ψ fixes all cycles in π of length other than l_i and rearranges and/or rotates the cycles of length l_i in all possible ways. By Lemma 3.3, therefore, $\langle \bigcup_{i=1}^r K_i \rangle = \mathscr{C}_{S_n}(Z)$.

Observe that $\sum_{i=1}^r m_i \cdot l_i = n$, thus $|K|$ is $O(n)$, and so K can be constructed in polynomial time. ▮

In Section 4, we shall show that $\mathscr{C}_{S_n}(Z)$ arises as the automorphism group of a very simple graph which can be constructed from π in $O(n)$ steps.

EXAMPLE 3.3. Let $\pi = (1, 2)(3, 5, 6)(4, 7)(8, 12)(9, 11, 13)(10)(14)$ be a permutation in S_{14} generating the group Z. Here π has two cycles of length 1, three cycles of length 2, and two cycles of length 3.

We consider first the cycles of length 3, which are $(3, 5, 6)$ and $(9, 11, 13)$. We obtain $\zeta_{1,1} = (3, 5, 6)$, $\zeta_{1,2} = (9, 11, 13)$, $\alpha_1 = (3, 9)(5, 11)(6, 13)$, and $\beta_1 = \alpha_1$. Thus, $K_1 = \{\zeta_{1,1}, \zeta_{1,2}, \alpha_1\}$. Next, we consider the three cycles of length 2, which are $(1, 2)$, $(4, 7)$, and $(8, 12)$. Here we obtain the set K_2 consisting of $\zeta_{2,1} = (1, 2)$, $\zeta_{2,2} = (4, 7)$, $\zeta_{2,3} = (8, 12)$, $\alpha_2 = (1, 4)(2, 7)$, and $\beta_2 = (1, 4, 8)(2, 7, 12)$. Finally, for the cycles of length 1, the permutations ζ are each the identity permutation, and $\alpha_3 = \beta_3 = (10, 14)$. Together, these permutations generate $\mathscr{C}_{S_n}(Z)$. ∎

THEOREM 3.5. *Problem* 3.7 *can be polynomial time reduced to Problem* 3.5.

Proof. Let $A, B < S_n$, with known generating sets, and assume we wish to find generators for $\mathscr{C}_A(B)$. We do so in two steps.

(a) Determine generators for $\mathscr{C}_{S_n}(B)$.

(b) Intersect $\mathscr{C}_{S_n}(B)$ with A.

In Section 4, we show how to do Step (a) in polynomial time. For the present, we do Step (a) by constructing $\mathscr{C}_{S_n}(\langle\beta\rangle)$ for each of the generators β of B, using Lemma 3.4, and then intersect these groups, obtaining $\mathscr{C}_{S_n}(B)$ by Lemma 3.3. This reduction can be done in polynomial time because of our implicit assumption that we are given a polynomial-sized generating set for the group B. ∎

THEOREM 3.6 (Luks). *Problem* 3.6 *can be reduced to Problem* 3.7 *in polynomial time.*

Proof. Let $A < S_n$, X a subset of $\{1,..., n\}$. We shall determine generators for A_X, the setwise stabilizer of X in A, using an algorithm for Problem 3.7.

Let A' be a group isomorphic to A acting on

$$Y = \{(i,j) \mid 1 \leqslant i \leqslant n, j = 1, 2\}$$

constructed by associating with $\alpha \in A$ the permutation α' of Y, where $(i,j)^{\alpha'} = (i^{\alpha},j)$.

Let π be the permutation of Y where, for $i \in X$, $(i,j)^{\pi} = (i,j)$, and, for $i \notin X$, $(i, 1)^{\pi} = (i, 2)$, $(i, 2)^{\pi} = (i, 1)$. Let Z be the group generated by π. Then it is easy to see that $\mathscr{C}_{A'}(Z)$, the centralizer in A' of Z, is isomorphic to A_X. Generators for A_X are easily obtained from generators for $\mathscr{C}_{A'}(Z)$. ∎

We have now established the polynomial time equivalence of Problems 3.4 through 3.7. A number of special cases of these problems are in **P**, and we discuss them in Section 4. There is also a special case of group intersection which is in **NP** $\cap$ co**NP**, [15].

3.3. *Isomorphism Complete Problems*

As pointed out already, graph isomorphism is a special case of the double coset membership problem. Since there are many publications on this problem, we can be very brief.

Problems polynomial time equivalent to graph isomorphism have been called *isomorphism complete.* There are many known isomorphism complete problems, e.g., [4, 23, 24]. One of these problems is

PROBLEM 3.8 (Graph Automorphism). Given a graph X with n vertices, determine a generating set for Aut(X), the automorphism group of the graph.

For a proof of the isomorphism completeness of Problem 3.8 see [23]. In [15], we give a proof that Aut(X) arises as the intersection of two permutation groups (of degree polynomial in n) for which small generating sets are known. This provides an alternate proof that Problem 3.8, graph isomorphism, and any other isomorphism complete problem can be polynomial time reduced to Problem 3.1. We have not found a reduction in the opposite direction.

3.4. *Remarks*

We have presented eight group-theoretic problems which are at least as hard as graph isomorphism. The problems are naturally related to graph isomorphism. For one, graph isomorphism can be reduced to each of these problems in polynomial time, by easy and straightforward reductions. Apparently the only previously known such reduction among natural problems in **NP** of unknown complexity status has been the reduction of group isomorphism to graph isomorphism [24]. Furthermore, as pointed out already, double coset membership is itself an abstract isomorphism question involving structures which are apparently more general than graphs.

It is only recently that the complexity of these group-theoretic problems has been studied. In view of this, our conjecture that we have here a part of a hierarchy between problems in **P** and **NP**-complete problems is not compelling. It does not seem out of the question that these problems ultimately belong to **P**. However, the length of time that graph isomorphism, the problem on the lowest level of difficulty, has resisted a polynomial time solution does not suggest this possibility.

It is also possible that double coset membership is an isomorphism complete problem (but does not have a polynomial time algorithm). After all, Problem 3.1 is essentially an isomorphism question. If, on the other hand, the problem remains harder than graph isomorphism, then this must mean that permutation groups of degree n are intrinsically more complicated than graphs of size polynomial in n. In particular, an open question closely related to the properness of the three levels in our hierarchy is the following:

PROBLEM. Given a permutation group G of degree n, is there a graph X with m vertices such that G is isomorphic to Aut(X), the automorphism group of X, and such that m is polynomial in n?

Frucht has shown that any permutation group G arises as the automorphism group of a graph X, [11]. The size of the graph X, however, is polynomial in the *order* of G, not the degree of G. Should the above question have an affirmative constructive answer, then it is not difficult to show that double coset membership is isomorphism complete.

4. Centralizer and Center

We now show that Problem 3.7, Centralizer in Another Group, has two special cases which are in **P**. Specifically, we show that we can efficiently find generators for the centralizer (in S_n) of any permutation group $G = \langle K \rangle$, for which we have a generating set K. Furthermore, observing that the centralizer of G also normalizes G, we show how to find generators for the center of G. The major part of this section describes the algorithm for the centralizer, since finding the center is an application of our intersection algorithm from [15].

We begin by reexamining the centralizer $\mathscr{C}_{S_n}(Z)$ of the cyclic group Z generated by a permutation π. Recall the proof of Lemma 3.4. We argued that ψ is in $\mathscr{C}_{S_n}(\langle \pi \rangle)$ iff conjugation under ψ rearranges the ordering of cycles of equal length in π and/or rotates the individual cycles. We now show that there is a directed graph X_π whose automorphism group is precisely $\mathscr{C}_{S_n}(\langle \pi \rangle) = \mathscr{C}_{S_n}(\pi)$.

Definition 4.1. A *cycle graph* is a directed graph $X = (V, E)$ such that, for every $v \in V$, the indegree and the outdegree of v are equal and are either 0 or 1.

Intuitively, a cycle graph consists of disjoint directed cycles and/or isolated points.

Definition 4.2. Let $\pi \in S_n$ be a permutation. The *cycle graph of* π, denoted X_π, is the cycle graph (V, E_π), where $V = \{1, \ldots, n\}$, and there is an edge from i to j in E_π if $i^\pi = j$.

Example 4.1. Let $\pi = (1, 2)(3, 5, 6)(4, 7)(8, 12)(9, 11, 13)(10)(14)$ be the permutation in S_{14} of Example 3.3. Then X_π is as shown in Fig. 4.1. ∎

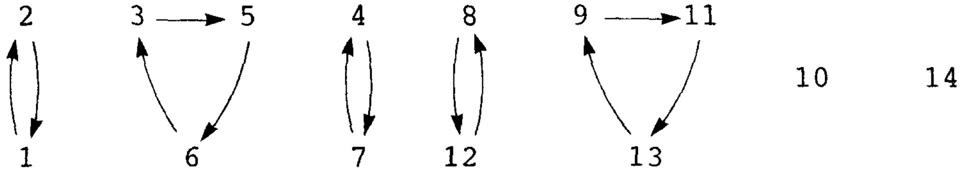

Figure 4.1

Note that X_π is unique, and that it can be constructed from π in $O(n)$ steps. From the proof of Lemma 3.4, the following is obvious:

LEMMA 4.1. *If $\pi \in S_n$ is a permutation and X_π is the cycle graph of π, then* $\mathrm{Aut}(X_\pi) = \mathscr{C}_{S_n}(\pi)$.

In Section 3, we constructed $\mathscr{C}_{S_n}(\langle K \rangle)$ by first constructing $\mathscr{C}_{S_n}(\pi)$ for all $\pi \in K$, and then intersecting these groups using Lemma 3.3. We observe now that instead of intersecting the groups $\mathscr{C}_{S_n}(\pi)$, $\pi \in K$, we can superimpose the graphs X_π resulting in a new graph X_K whose automorphism group will be $\mathscr{C}_{S_n}(\langle K \rangle)$. Here we need to color the edges of each graph X_π uniformly to prevent an automorphism of X_K from mapping an edge belonging to X_π to an edge belonging to X_ψ, where $\pi \neq \psi$.

DEFINITION 4.3. A *colored cycle graph* X of degree k is a directed multigraph (V, E) and a mapping from E into the set $\{c_1, ..., c_k\}$ such that the subgraph obtained by deleting all edges in E which are not mapped into c_i is a cycle graph $(1 \leqslant i \leqslant k)$.

Intuitively, a colored cycle graph is obtained by superimposing k cycle graphs, where the edges of the ith cycle graph are colored c_i. Note that the resulting graph is a multigraph, since there may be more than one edge from a vertex v to a vertex w. Note, however, that two edges from v to w must be of different color. The individual monochromatic cycle graphs are called the *color constituents* of X.

DEFINITION 4.4. Let K be a set of permutations in S_n. The *colored cycle graph X_K of K* is the colored cycle graph whose color constituents are the cycle graphs X_π, $\pi \in K$.

Thus, X_K is of degree $|K|$. Observe that we can construct X_K from K in $O(n \cdot |K|)$ steps.

EXAMPLE 4.2. Let $G = \langle K \rangle$, where $K = \{(1, 2, 3, 4), (2, 4)\}$. Here $G = D_4$, the dihedral group of degree 4. The colored cycle graph X_K of K is shown in Fig. 4.2. ▮

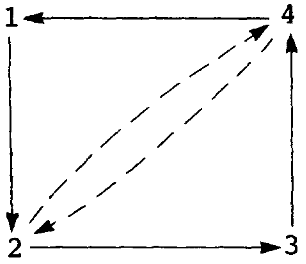

FIGURE 4.2

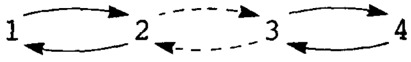

FIGURE 4.3

Because of the coloring of X_K and by Lemma 3.3, the following is obvious:

LEMMA 4.2. *Let $G = \langle K \rangle$, K a subset of S_n. Then $\mathscr{C}_{S_n}(G) = \mathrm{Aut}(X_K)$, where X_K is the cycle graph of K.*

We shall show how to test isomorphism and determine the automorphism group of colored cycle graphs.

DEFINITION 4.5. Let X be a connected colored cycle graph, x and y distinct vertices of X. The *cycle distance* $d(x, y)$ of x and y is defined by

(1) $d(x, y) = 0$ iff x and y lie on a common cycle in X whose edges are of uniform color c_i.

(2) $d(x, y) = k$ iff $z_1, ..., z_k$ is the smallest number of vertices in X such that $d(x, z_1) = 0$, $d(z_k, y) = 0$, and $d(z_i, z_{i+1}) = 0$, $1 \leqslant i < k$.

EXAMPLE 4.3. Let X be the connected colored cycle graph shown in Fig. 4.3. Then $d(1, 2) = 0$, $d(1, 3) = 1$, $d(1, 4) = 2$. ∎

The isomorphism test of colored cycle graphs to be described next rests on the following key observation:

THEOREM 4.1. *Let $X = (V_X, E_X)$ and $Y = (V_Y, E_Y)$ be isomorphic connected colored cycle graphs. Then every isomorphism map ι from X to Y is fully determined by the image x^ι of an arbitrary vertex $x \in V_X$.*

Proof. Without loss of generality, we assume that both X and Y have more than one vertex. So, let ι be an isomorphism map from X to Y, x any vertex in X, and x^ι the image of x under ι in Y. Since monochromatic cycles of color c_i are mapped again into monochromatic cycles of the same color in Y, and since cycles in every color constituent are disjoint, it follows that the image of every vertex u in X of cycle distance 0 from x is determined by x^ι. Since vertices z in X of cycle distance 1 from x have a cycle distance 0 from some other vertex u in X with $d(x, u) = 0$, the same argument shows that the image of every such z is fully determined by u^ι and therefore by x^ι. Proceeding by induction on the cycle distance from x, since both X and Y are connected, it follows that the image of every vertex of X under ι is fully determined by x^ι. ∎

From Theorem 4.1, we obtain Corollaries 4.1–4.3.

COROLLARY 4.1. *If X is a connected colored cycle graph with n vertices, then the order of* $\mathrm{Aut}(X)$ *is at most n.*

Proof. Any automorphism is an isomorphism from X to X. ▮

In particular, the stabilizer of any vertex in $\mathrm{Aut}(X)$ must be the trivial group. Therefore, if $\alpha \in \mathrm{Aut}(X)$, then the cycles in α are all of equal length.

COROLLARY 4.2. *If X and Y are connected colored cycle graphs of degree k with n vertices each, then isomorphism of X and Y can be tested in* $O(n^2 \cdot k)$ *steps.*

Proof. Let x be a fixed arbitrary vertex of X. In $O(n \cdot k)$ steps we can test, for each vertex y in Y, whether mapping x into y determines an isomorphism. ▮

COROLLARY 4.3. *If X is a connected colored cycle graph of degree k with n vertices, then generators for* $\mathrm{Aut}(X)$ *can be determined in* $O(n^2 \cdot k)$ *steps.*

Proof. Immediate from Corollaries 4.1 and 4.2. ▮

We show how to test isomorphism of colored cycle graphs which are not necessarily connected. Intuitively, we split the graphs X and Y into components, and test isomorphism of each component. Having classified the components into isomorphism classes, it is clear that X and Y are isomorphic iff exactly half the components in each isomorphism class belong to X.

THEOREM 4.2. *Let X and Y be colored cycle graphs of degree k with n vertices each. Then we can test isomorphism of X and Y in* $O(n^2 \cdot k)$ *steps.*

Proof. In $O(n \cdot k)$ steps we can find the components of X and Y. If X and Y do not have an equal number of components of equal size, then they cannot be isomorphic. So, let X and Y each have p_i components of size m_i, $1 \leqslant i \leqslant r$. We test at most p_i^2 components of size m_i for isomorphism. By Corollary 4.2, this requires a total of $O(p_i^2 \cdot m_i^2 \cdot k)$ steps. Since $\sum_{i=1}^{r} p_i \cdot m_i = n$, isomorphism of X and Y can be tested in $O(n^2 \cdot k)$ steps. It is clear that we can construct an isomorphism map in the same time bound. ▮

We therefore obtain

COROLLARY 4.4. *If X is a colored cycle graph of degree k with n vertices, then generators for* $\mathrm{Aut}(X)$ *can be found in* $O(n^2 \cdot k)$ *steps.*

Proof. $\mathrm{Aut}(X)$ is generated by a set $K_1 \cup K_2$, where K_1 generates all automorphisms which fix setwise the vertices of each component of X, and K_2 generates all possible permutations of isomorphic components of X.

Let X have p_i components of size m_i, $1 \leqslant i \leqslant r$. We can find these components in $O(n \cdot k)$ steps. By Corollary 4.3, we find the set K_1 in $O((\sum_{i=1}^{r} p_i \cdot m_i^2) \cdot k)$ steps, which is dominated by $O(n^2 \cdot k)$.

By Theorem 4.2, we can classify the components of X into isomorphism classes in

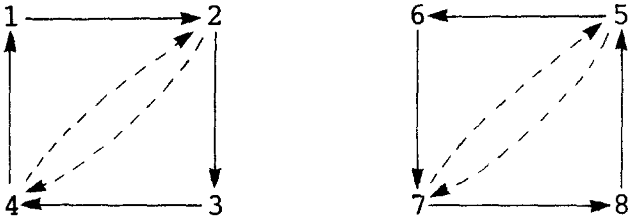

FIGURE 4.4

$O(n^2 \cdot k)$ steps, and can find, in every isomorphism class, an isomorphism map from an arbitrary representative component to each component in the class. It is trivial to produce the set K_2 from these maps.

Note that both $|K_1|$ and $|K_2|$ are $O(n)$. ▮

EXAMPLE 4.4. Let $K = \{(1, 2, 3, 4)(5, 6, 7, 8), (2, 4)(5, 7)\}$. The colored cycle graph X_K is of degree 2 and is shown in Fig. 4.4. The graph X_K has two isomorphic components with $\iota = (1, 6)(2, 7)(3, 8)(4, 5)$ establishing the isomorphism. Note that ι is completely determined by $(1, 6)$. Furthermore, the automorphisms stabilizing setwise the component vertices are generated by $\alpha = (1, 3)(2, 4)$ and $\beta = (6, 8)(5, 7)$. Thus, $\mathrm{Aut}(X_K) = \langle \iota, \alpha, \beta \rangle$. ▮

COROLLARY 4.5. *Let K be a set of permutations in S_n generating a group G. Then generators for $\mathscr{C}_{S_n}(G)$ can be found in $O(n^2 \cdot |K|)$ steps.*

Proof. This result is immediate from Corollary 4.4 and Lemma 4.2. ▮

We can exploit the geometric presentation of $\mathscr{C}_{S_n}(G)$ to prove directly a few results about the structure of centralizers. We state these results informally, since they follow so easily from the representation of $\mathscr{C}_{S_n}(\langle K \rangle)$ as automorphism group of the graph X_K.

We split X_K into its components, which we then classify into the isomorphism classes $J_1, \ldots, J_s$. Since components in distinct classes are not isomorphic, it follows that $\mathrm{Aut}(X_K)$ and therefore $\mathscr{C}_{S_n}(\langle K \rangle)$ are the direct product

$$W_1 \times W_2 \times \cdots \times W_s$$

of certain groups W_i which act, respectively, on the vertices of the components in the class J_i.

Next, let J_i consist of m_i components. Since these components are isomorphic graphs, they can be permuted, setwise, in all $m_i!$ ways. Furthermore, since the automorphism groups of isomorphic graphs are isomorphic, it follows that the group W_i is isomorphic to the wreath product of a group G_i by the symmetric group S_{m_i}. Here the group G_i is the automorphism group of a representative component in the class J_i.

The groups G_i are, by Corollary 4.2, of order not exceeding n_i, where n_i is the number of vertices of each component in the class J_i. Thus, the order of W_i is at most $n_i^{m_i} \cdot (m_i!)$, thus $|\mathscr{C}_{S_n}(G)| \leqslant \prod_{i=1}^{s}(n_i^{m_i} \cdot (m_i!))$. Note that $\sum_{i=1}^{s} n_i \cdot m_i = n$. Furthermore, by Theorem 4.1, the groups G_i contain only permutations α such that α consists of cycles of equal length. Consequently, the group G_i is isomorphic to each of its transitive constituents, which are, in turn, regular groups.

We remark without proof that in light of these structural results it is easy to modify the method of Corollary 4.4, so as to determine a strong generating set for $\mathscr{C}_{S_n}(G)$, in the same time bound (cf. [28]). The significance of this is that a strong generating set permits an $O(n^2)$ membership test in the generated group. For other generating sets, the membership test requires a preprocessing computation of cost $O(n^6)$ steps. Thus, it appears that determining the centralizer of $\langle K \rangle$ is computationally less expensive than determining $\langle K \rangle$ itself.

We conclude by showing how to use the centralizer algorithm for determining generators for the center of a permutation group.

LEMMA 4.3. *Let $G < S_n$ be a permutation group. Then $\mathscr{C}_{S_n}(G)$ normalizes G, i.e., for all $\pi \in \mathscr{C}_{S_n}(G)$, $\pi G = G\pi$.*

The lemma is obvious from the definition of $\mathscr{C}_{S_n}(G)$. Applying a result from [15], we obtain

THEOREM 4.3. *Let $G < S_n$ be a permutation group of degree n generated by a set K of permutations. Then we can determine a generating set for $\mathscr{C}_G(G)$, the center of G, in $O(n^2 \cdot |K| + n^6)$ steps.*

Proof. Using Corollary 4.5, we construct a generating set K' of size $O(n^2)$ for $\mathscr{C}_{S_n}(G)$ in $O(n^2 \cdot |K|)$ steps. By Lemma 4.3, we can intersect $\mathscr{C}_{S_n}(G)$ with G using our $O(n^6)$ algorithm for intersecting with a normalizing group. ∎

ACKNOWLEDGMENT

It is a pleasure to acknowledge the free discussions and correspondence with E. Luks, which greatly contributed towards focusing this material.

REFERENCES

1. L. BABAI, Monte Carlo algorithms in graph isomorphism testing, *SIAM J. Comput.* (1979), submitted.
2. L. BABAI, Y. GRIGORYEV, AND D. MOUNT, Isomorphism testing for graphs with bounded eigenvalue multiplicities, *in* "Proceedings, 14th STOC Symposium," San Francisco, 1982.
3. K. BOOTH, Isomorphism testing for graphs, semigroups, and finite automata are polynomially equivalent problems, *SIAM J. Comput.* 7 (3) (1978), 273–279.
4. K. BOOTH AND C. COLBOURN, "Problems Polynomially Equivalent to Graph Isomorphism," Tech. Report CS-77-04, Department of Computer Sciences, University of Waterloo, Canada, June 1979.

5. H. Brown, L. Hjelmeland, and L. Masinter, Constructive graph labelling using double cosets, *Discrete Math.* **7** (1974), 1–30.

6. G. Butler, "Computational Approaches to Certain Problems in the Theory of Finite Groups," Ph.D. Dissertation, Department of Mathematics, Univ. of Sydney, Australia, 1979.

7. M. Furst, J. Hopcroft, and E. Luks, "A Subexponential Algorithm for Trivalent Graph Isomorphism," Tech. Report 80-426, Computer Science Department, Cornell Univ., Ithaca, N.Y., 1980.

8. M. Furst, J. Hopcroft, and E. Luks, Polynomial time algorithms for permutation groups, *in* "Proceedings, 21st STOC," pp. 36–41, Syracuse, N.Y., 1980.

9. I. Filotti, G. Miller, and J. Reif, On determining the genus of a graph in $O(V^{O(g)})$ steps, *in* "Proceedings, 11th Annual STOC Symposium," pp. 27–37, 1979.

10. I. Filotti and J. Mayer, A polynomial time algorithm for determining isomorphism of graphs of fixed genus, *in* "Proceedings, 12th Annual STOC Symposium," pp. 236–243, 1980.

11. R. Frucht, Herstellung von Graphen mit vorgegebener abstrakter Gruppe, *Compositio Math.* **6** (1938), 239–250.

12. M. Garey and D. Johnson, "Computers and Intractability, a Guide to the Theory of **NP**-Completeness," Freeman, San Francisco, 1979.

13. M. Hall, Jr., "The Theory of Groups," Macmillan, New York, 1959.

14. C. Hoffmann, Testing isomorphism of cone graphs, *in* "Proceedings, 12th Annual STOC Symposium," pp. 244–251, 1980.

15. C. Hoffmann, "On the Complexity of Intersecting Permutation Groups and its Relationship with Graph Isomorphism," Tech. Report 4/80, Dept. of Informatik, University of Kiel, West Germany, 1980.

16. J. Hopcroft and R. Tarjan, A $V \log V$ algorithm for isomorphism of triconnected planar graphs, *J. Comput. System Sci.* **7** (1973), 323–331.

17. J. Hopcroft and J. Wong, A linear time algorithm for isomorphism of planar graphs, *in* "Proceedings, 6th Annual STOC Symposium," pp. 172–184, 1974.

18. R. Ladner, On the structure of polynomial time reducibility, *J. Assoc. Comput. Mach.* **22** (1975), 155–171.

19. D. Lichtenstein, Isomorphism for graphs embeddable on the projective plane, *in* "Proceedings, 12th Annual STOC Symposium," pp. 218–224, 1980.

20. A. Lubiw, Some **NP**-complete problems similar to graph isomorphism, *SIAM J. Comput.* **10** (1981), 11–21.

21. E. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *in* "Proceedings, 21st Annual FOCS Symposium," pp. 42–49, 1980.

22. E. Luks, "The Complexity of Permutation Group Problems," presented at the summer meeting of the AMS, Pittsburgh, 1981.

23. R. Mathon, A note on the graph isomorphism counting problem, *Inform. Process. Lett.* **8** (1979), 131–132.

24. G. Miller, Graph isomorphism, general remarks, *in* "Proceedings, 9th STOC," pp. 143–150, 1977.

25. G. Miller, On the $n^{\log n}$ isomorphism technique, *in* "Proceedings, 10th Annual STOC Symposium," pp. 51–58, 1978.

26. G. Miller, Isomorphism testing for graphs of bounded genus, *in* "Proceedings, 12th Annual STOC Symposium," pp. 218–224, 1980.

27. C. Sims, Determining the onjugacy classes of a permutation group, *in* "Proceedings of the Symposium on Applied Mathematics" (G. Birkhoff and M. Hall, Eds.), 191–195, New York, 1970.

28. C. Sims, Computation with permutation groups, *in* "Proceedings, 2nd ACM Symposium on Symbolic and Algebraic Manipulations," pp. 23–28, Los Angeles, 1971.

29. L. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* **8** (1979), 189–202.

30. H. Wielandt, "Finite Permutation Groups," Academic Press, New York, 1964.

31. M. Fontet, Calcul de Centralisateur d'un Groupe de Permutations, *Bull. Soc. Math. France Mem.* **49–50** (1977), 53–63.