Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 25, No. 1, August 1982
Printed in Belgium

Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time*

EUGENE M. LUKS

Department of Mathematics, Bucknell University, Lewisburg, Pennsylvania 17837

Received October 21, 1981

Suppose we are given a set of generators for a group G of permutations of a colored set A. The color automorphism problem for G involves finding generators for the subgroup of G which stabilizes the color classes. It is shown that testing isomorphism of graphs of bounded valance is polynomial-time reducible to the color automorphism problem for groups with composition factors of bounded order. The algorithm for the latter problem involves three divide-and-conquer maneuvers. The problem is solved sequentially on the G-orbits. An orbit is broken into a minimal system of imprimitivity blocks. At that point, the hypothesis on G guarantees the presence of a subgroup P of "small" index which acts a p-group on the blocks. Divide-and-conquer is then used on the group, trading the problem on G for a small number of similar problems on P. In the trivalent case $p = 2$, $P = G$ and the analysis requires only elementary notions. For higher valence, the justification requires some new observations about primitive permutation groups.

INTRODUCTION

It is well known that testing isomorphism of graphs is polynomial-time reducible to the problem of finding a set of generators for the group, $\mathrm{Aut}(X)$, of automorphisms of a graph X. We recall the reduction:

We may suppose that we wish to compare two *connected* graphs X_1, X_2. Form the disjoint union $X = X_1 \cup X_2$. Then X_1 and X_2 are isomorphic if and only if an automorphism of X switches the two connected components. Furthermore, if such automorphisms of X exist, at least one must turn up in any set of generators of $\mathrm{Aut}(X)$.

The problem of determining generators for $\mathrm{Aut}(X)$ is, in turn, reducible in several ways to inherently algebraic questions. In this paper, we shall deal with the

Color Automorphism Problem. Input: A colored set A and generators for a group G of permutations of A. *Find*: Generators for the subgroup consisting of the color preserving maps.

Computing $\mathrm{Aut}(X)$ is a special case: Let G be the group of all permutations of the

* This research was supported in part by National Science Foundation Grant MCS-8102856.

0022-0000/82/040042-24$02.00/0

vertex set $\mathcal{V}(X)$ but view G as acting on the set A of unordered pairs of vertices; color A with two colors to delimit edges and non-edges of X; then $\mathrm{Aut}(X)$ is the color-preserving subgroup. Notice that, for any graph X, we arrive at a color automorphism problem with G a complete symmetric group S_n (acting on an $\binom{n}{2}$element set). We shall see that, for special classes of graphs, there are independent reductions to color automorphism problems involving other groups. In fact, it is crucial to the timing of our fundamental algorithm that the group have a property (specifically, small composition factors) not shared by S_n.

Though such reductions of the problem have long been available, few papers on isomorphism testing have utilized, in a substantive way, the machinery, both theoretical and computational, of permutation groups. Thus, an important breakthrough occurs in the recent work of Babai [2]. He considered vertex-colored graphs with bounded color classes and described a polynomial-time probabilistic $(R \cap coR)$ algorithm for computing $\mathrm{Aut}(X)$. The algorithm uses, in a significant manner, the fact that $\mathrm{Aut}(X)$ is contained in a given direct product of small groups. Babai's success inspired a closer look at permutation group algorithms and their relation to graph isomorphism by Furst *et al.* [8, 9]. In particular, it was shown that Babai's methods could be made deterministic with no essential loss of efficiency. In fact, similar algorithms already lay (unanalyzed) in the computational-group-theory literature.

We remark that soon after Babai's announcement, Hoffman [11] described an algorithm for "cone graphs." The graphs were designed to admit a recursive application of Babai's methods and it was claimed that the isomorphism test required only $n^{\log n}$ time. This, in turn, led to an announcement of an $n^{\log n}$ algorithm for trivalent graphs by Furst *et al.* [8]. The latter report reduced trivalent-graph isomorphism to cone-graph isomorphism using the natural binary tree structure of Sylow 2-subgroups of S_n. Though the reduction remains intact, there seems to be a gap in Hoffman's analysis which would invalidate the $n^{\log n}$ claim for the algorithm in [8] as well.

The present paper begins a deeper probe into the underlying group theory. We present some new algebra as well as some new algorithms. Isomorphism testing of graphs of valence $\leqslant t$ is reduced to the color automorphism problem for groups whose composition factors are subgroups of S_{t-1}. For this class of groups two naive (though previously overlooked) divide-and-conquer tricks are introduced on the underlying set. It is then an elementary exercise to exhibit a polynomial time-bound in the trivalent case. The key fact is that the divide-and-conquer only gets "hung up" when it is faced with a primitive group. However, in the trivalent case, the groups are 2-groups and primitive 2-groups can only have order 2. Well, more generally, primitive p-groups can only have order p. This implies the color automorphism algorithm is actually uniformly efficient over p-groups. That phenomenon is exploited to speed up the process for higher valence. Although the primitive groups that arise are not p-groups, they are almost so. To be precise, we show that they have p-subgroups of polynomial index and that such subgroups can be located in polynomial time. These observations form the core of the extension to higher valence. Thus, a

third divide-and-conquer strategem is introduced, breaking the problem into a small number of similar problems for p-groups.

We introduce terminology and recall some basic facts and algorithms in Section 1. In Section 2, we describe the algorithm as it applies to trivalent graphs. The extension to graphs of bounded valence is described in Section 3. The group-theoretic justification of the procedure is presented in subsection 3.2. We conclude in Section 4 with some remarks concerning extensions, other applications, and open problems.

Finally, we remark that, as is clear to those familiar with the literature, the major result right now is that this problem is in P. Thus, we avoid the unnecessary complications that would result from an attempt to justify precise upper bounds. In fact, we do not always present our "best" algorithms (see 4.1).

1. Preliminaries

1.1. *Notations and Background*

For a graph X, $\mathcal{V}(X)$ denotes the set of vertices, $\mathcal{E}(X)$ the set of edges, $\langle v, w\rangle$ denotes an edge joining vertices v and w, $\mathrm{Aut}(X)$ denotes the group of automorphisms of X, $\mathrm{Aut}_e(X)$ the subgroup fixing the edge e.

The group of permutations of an n-element set is denoted by S_n or, if the set requires explication, by $\mathrm{Sym}(A)$. A subset G of $\mathrm{Sym}(A)$ is said to *stabilize* $B \subseteq A$ if $\sigma(B) = B$ for $\sigma \in G$. In some instances we refer to the *action* of a group G on a set B; that is, we suppose only that there is a homomorphism $G \to \mathrm{Sym}(B)$. Such actions arise quite naturally in this paper since, given $G \subseteq \mathrm{Sym}(A)$, we shall often consider the induced action on G-stable subsets of A and on collections of subsets of A. The action of G on B is called *faithful* if the homomorphism $G \to \mathrm{Sym}(B)$ is injective. If G acts on B and $b \in B$, the *G-orbit* of b is the set $\{\sigma(b) \mid \sigma \in G\}$; we say G acts *transitively* on B if B is a G-orbit. Let G be a group acting transitively on a set A. A *G-block* is a subset B of A, $B \neq \varnothing$ or A, such that, for all $\sigma, \tau \in G$, $\sigma(B) = \tau(B)$ or $\sigma(B) \cap \tau(B) = \varnothing$. (We depart from tradition in not insisting that $|B| > 1$.) If B is a G-block we call the collection $\{\sigma(B) \mid \sigma \in G\}$ a *G-block system in A*. The group G then acts transitively on the blocks of the system. We say that G acts *primitively* on A (or if $G \subseteq \mathrm{Sym}(A)$ we call G a *primitive* group) if there are no G-blocks of size >1. A G-block system is said to be *minimal* if G acts primitively on the blocks. (Note: It is the *number* of blocks that is minimal.) The number of blocks in a minimal G-block system is not, in general, uniquely determined. However, one knows

Lemma 1.1. *Let P be a transitive p-subgroup of* $\mathrm{Sym}(A)$ *with* $|A| > 1$. *Then any minimal p-block system consists of exactly p blocks. Furthermore, the subgroup P' which stabilizes all of the blocks has index p in P.*

Proof. The quotient P/P' is a primitive p-group (acting on the blocks) and so the order of P/P' = the number of blocks = p [10, p. 66]. ∎

For further background on permutation groups, we refer the reader to [12] or [26].

If Φ is a subset of the group G, $\langle \Phi \rangle$ denotes the subgroup of G generated by Φ. If H is a subgroup of G then $|G:H|$ denotes the index of H in G. We denote the identity element of a group and the trivial subgroup it constitutes by 1. We write $N \lhd G$ if N is a normal subgroup of G. A *composition series* for a group G is a chain of subgroups of the form

$$1 = G^m \lhd \cdots \lhd G^2 \lhd G^1 \lhd G^0 = G,$$

in which the quotients G^i/G^{i+1} are simple groups. By the Jordan–Hölder Theorem [10, 12] the collection of quotient groups is independent of the choice of composition series. The groups in this collection are called the *composition factors* of G.

1.2. *Some Basic Algorithms*

Since any group G has a generating set of cardinality $\log |G|$ or less, subgroups of S_n can be specified in space which is polynomial in n. The succinctness of such presentations raises the issue of whether fundamental questions about the group can be answered in polynomial time. The quest for *efficient* techniques for handling large permutation groups has, of course, a long history (see [24]). Still, complexity analyses of the algorithms have only recently appeared. In [9], the author, jointly with Furst and Hopcroft, demonstrated that several basic computational problems have polynomial-time solutions. The basic tool (the subgroup chain $\{G_i\}$), however, is apparently due to Sims [23, 24] and has been in use for some time. We extract from [9] the following

LEMMA 1.2 [Furst–Hopcroft–Luks]. *Given a set of generators for a subgroup G of S_n one can determine in polynomial-time*

(i) *the order of G;*

(ii) *whether a given permutation σ is in G;*

(iii) *generators for any subgroup of G which is known to have polynomially bounded index in G and for which a polynomial-time membership test is advailable.*

For the reader's convenience we review the algorithms for Lemma 1.2. Suppose G is a subgroup of $\mathrm{Sym}(A)$, where $A = \{a_1, \ldots, a_n\}$. Denote by G_i the subgroup of G which fixes all of the points in $\{a_1, \ldots, a_i\}$. Thus we have a chain of subgroups

$$1 = G_{n-1} \subseteq \cdots \subseteq G_1 \subseteq G_0 = G.$$

The algorithm for Lemma 1.2(i) involves a simultaneous construction of complete sets of coset representatives, C_i, for G_i modulo G_{i+1}, $0 \leqslant i \leqslant n-2$. Then $|G|$ is the product $|C_0| \cdot |C_1| \cdots |C_{n-2}|$. The building block in this construction is the following subroutine. The input is an element $\alpha \in G$. The lists C_i contain (not necessarily complete) sets of left coset representatives for G_i modulo G_{i+1}.

```
procedure Filter (α)
  for i = 0 until n − 2 do
    begin
      if γ^{-1}α ∈ G_{i+1} for some γ ∈ C_i
        then α ← γ^{-1}α
        else add α to C_i; return
    end
  return
```

Thus the subroutine searches for a representative of the coset of α modulo G_1 in the list C_0. If it is not found, then α represents a previously undiscovered coset and it is added to the list. If it is found in the guise of γ then $\gamma^{-1}\alpha$ is in G_1 and its class modulo G_2 is sought in C_1, etc. Since, for $\sigma \in G_i$, membership in G_{i+1} is testable in constant time, the procedure requires only polynomial time. Observe that the result of calling Filter for $\alpha \in G$ is that (the original) α is in $C_0 C_1 \cdots C_{n-2}$. This holds whether α caused an increase in some C_i or whether it "survived" the filtration.

The algorithm for Lemma 1.2(i) is now easily stated:

(1) Initialize $C_i \leftarrow \{1\}$ for all i.

(2) Filter the set of generators of G.

(3) Filter the sets $C_i C_j$ with $i \geqslant j$.

Of course, calls to the subroutine may result in an increase in some C_i, thus demanding more runs of (3). However, we know a priori that, at any stage, $|C_i| \leqslant |G_i : G_{i+1}| \leqslant n - i$. Thus the process terminates in polynomial time. The result of (2) is that the original generating set is contained in $C_0 C_1 \cdots C_{n-2}$. The actual outcome of (3), given (1), is that $C_i C_j \subseteq C_j C_{j+1} \cdots C_{n-2}$. These facts can be used to prove [9] that $G = C_0 C_1 \cdots C_{n-2}$. That C_i represents G_i modulo G_{i+1} is then immediate.

Given Lemma 1.2(i), an algorithm for Lemma 1.2(ii) is an immediate consequence of the observation: $\sigma \in \langle \Phi \rangle$ if and only if $|\langle \Phi, \sigma \rangle| = |\langle \Phi \rangle|$. Digging a little deeper, we observe that this membership test might be implemented by a construction of the lists $\{C_i\}$ for $\langle \Phi \rangle$ followed by the call Filter (σ). Then σ is in G if and only if it survives the filtration (i.e., it doesn't force an increase in some C_i).

For Lemma 1.2(iii), we alter the group chain to

$$1 = H_{n-1} \subseteq \cdots \subseteq H_2 \subseteq H_1 \subseteq H \subseteq G$$

and apply the same algorithm to generate complete sets of coset representatives. Note that the polynomial index of H in G and the requirement that membership in H be polynomially decidable guarantees again that the entire process takes only polynomial time. Ignoring the first list, namely the representatives of G modulo H, the remaining lists comprise a set of generators for H. (Another description of the essence of this algorithm can be modeled after the proof of Proposition 3.10.)

Given generators for $G \subseteq \mathrm{Sym}(A)$ it is an easy matter to determine the G-orbits using a transitive closure algorithm and this process will be used routinely. We shall

also need, in the transitive case, to be able to decompose the set further relative to the group action, namely, into non-trivial blocks of imprimitivity (if such exist). We observe now that this goal, too, is achievable in polynomial time. To be precise, we fix $a \in A$ and for each $b \in A$, $b \neq a$, we generate the (unique) smallest G-block containing $\{a, b\}$. As Sims has observed [22], this is precisely the connected component of a in the graph X with $\mathcal{V}(X) = A$ and $\mathcal{E}(X) =$ the G-orbit of $\{a, b\}$ in the set of all (unordered) pairs of elements of A. If G is imprimitive, the block must be proper for some choice of b. In that case, the connected components of X define a G-block system. Then, repeating the process, as necessary, with the induced action of G on the blocks, we actually have an algorithm for

LEMMA 1.3. *Given a set of generators for a subgroup G of S_n and a G-orbit B, one can determine, in polynomial time, a minimal G-block system in B.*

We remark that Atkinson [1] has described a particularly efficient implementation of the above ideas. In our applications it will be necessary to determine, as well, the subgroup of G which stabilizes all of the blocks.

LEMMA 1.4. *Let G, B be as above. Generators for the subgroup of G which stabilizes all of the blocks in a G-block system in B can be found in polynomial time.*

Lemma 1.2(iii), for example, guarantees this. Let $G_{(i)}$ denote the subgroup which stabilizes each of the first i blocks. Then (taking $G = G_{(0)}$)

$$[G_{(i)} : G_{(i+1)}] \leqslant \text{number of blocks} - i.$$

2. THE TRIVALENT CASE

2.1. *Reduction to the Color Automorphism Problem*

We demonstrate that the problem of testing isomorphism of trivalent graphs is polynomial-time reducible to the Color Automorphism Problem for 2-groups. The first step is a modification of the reduction to an automorphism problem. The motivation is Tutte's observation (Proposition 2.2) that $\mathrm{Aut}_e(X)$ is a 2-group.

PROPOSITION 2.1. *Testing isomorphism of trivalent graphs is polynomial-time reducible to the problem of determining generators for $\mathrm{Aut}_e(X)$, where X is a connected trivalent graph and e is a distinguished edge.*

Proof. Assume we possess a polynomial-time algorithm which returns generators for any such $\mathrm{Aut}_e(X)$. Once again, it suffices to be able to compare two connected trivalent graphs X_1, X_2. Fix an edge $e_1 \in \mathcal{E}(X_1)$. For each edge $e_2 \in \mathcal{E}(X_2)$ we can test whether there is an isomorphism from X_1 to X_2 which maps e_1 to e_2 as follows: Construct a connected trivalent graph X from the disjoint union $X_1 \cup X_2$ by (i) inserting new vertices v_1 in e_1 and v_2 in e_2, and (ii) joining v_1 to v_2 with a new edge e.

Then there is an isomorphism from X_1 to X_2 mapping e_1 to e_2 if and only if some element of $\mathrm{Aut}_e(X)$ transposes v_1 and v_2. Furthermore, if such automorphisms exist, any set of generators of $\mathrm{Aut}_e(X)$ will contain one. ∎

We now fix a connected trivalent graph X with $|\mathscr{V}(X)| = n$. The group $\mathrm{Aut}_e(X)$ is determined through a natural sequence of successive "approximations," $\mathrm{Aut}_e(X_r)$, $r = 1, 2, \ldots$, where X_r is the subgraph consisting of all vertices and all edges of X which appear in paths of length $\leqslant r$ through e. So X_1 is e itself and $X_{n-1} = X$. The groups are related via the induced homomorphisms

$$\pi_r: \mathrm{Aut}_e(X_{r+1}) \to \mathrm{Aut}_e(X_r)$$

in which $\pi_r(\sigma)$ is the restriction of σ to X_r. Thus, assuming we know $\mathrm{Aut}_e(X_r)$, the determination of $\mathrm{Aut}_e(X_{r+1})$ breaks up into two problems:

(I) Find a set, $\mathscr{R}$, of generators for K_r, the kernel of π_r.

(II) Find a set, $\mathscr{S}$, of generators for $\pi_r(\mathrm{Aut}(X_{r+1}))$, the image of π_r.

Then, if $\mathscr{S}'$ is any pullback of $\mathscr{S}$ in $\mathrm{Aut}_e(X_{r+1})$ (i.e., $\pi_r(\mathscr{S}') = \mathscr{S}$), $\mathscr{R} \cup \mathscr{S}'$ generates $\mathrm{Aut}_e(X_{r+1})$.

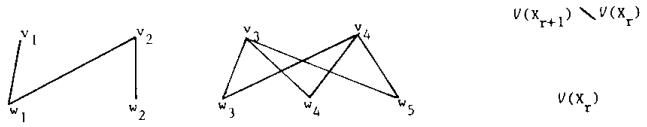
We shall see that the essential, and difficult, problem is (II). We remark that it is this problem that was reduced to cone graph isomorphism in [8].

To investigate these problems we consider $\mathscr{V}(X_{r+1}) \setminus \mathscr{V}(X_r)$. Each vertex in this set is connected to one, two or three vertices in X_r. We codify this relationship as follows: Let A denote the collection of all subsets of $\mathscr{V}(X_r)$ of size one, two, or three. Define

$$f: \mathscr{V}(X_{r+1}) \setminus \mathscr{V}(X_r) \to A$$

by $f(v) = \{w \in \mathscr{V}(X_r) \mid \langle v, w \rangle \in \mathscr{E}(X)\}$.

EXAMPLE.



Here

$$f(v_1) = \{w_1\}, \qquad f(v_2) = \{w_1, w_2\}, \qquad f(v_3) = f(v_4) = \{w_3, w_4, w_5\}.$$

We call v, v', for $v \neq v'$, *twins* if $f(v) = f(v')$ (Note: Triplets cannot exist). In the above example v_3 and v_4 are twins, v_1 and v_2 are not. Now,

$$\sigma \in \mathrm{Aut}_e(X_{r+1}) \qquad \text{implies} \qquad f(\sigma(v)) = \sigma(f(v)). \tag{*}$$

Thus, in particular, if $\sigma \in K_r$ (i.e., σ fixes all elements of X_r) then $f(v) = f(\sigma(v))$; so

either $v = \sigma(v)$ or v and $\sigma(v)$ are twins. It follows that K_r is precisely the elementary abelian 2-group generated by the transpositions in each pair of twins.

Since $|\mathrm{Aut}_e(X_{r+1})| = |\mathrm{Image}\ \Pi_r| \cdot |K_r|$, an induction argument recovers

PROPOSITION 2.2 (Tutte). *For each r, $\mathrm{Aut}_e(X_r)$ is a 2-group.*

To get at (II), observe that (*) implies any $\sigma \in \pi_r(\mathrm{Aut}_e(X_{r+1}))$ stabilizes the set of fathers with one son, i.e.,

$$A_1 = \{a \in A \mid a = f(v) \text{ for some unique } v \in \mathscr{V}(X_{r+1}) \backslash \mathscr{V}(X_r)\}.$$

Furthermore any $\sigma \in \pi_r(\mathrm{Aut}_e(X_{r+1}))$ must stabilize the subset of A consisting of the fathers of twins, i.e.,

$$A_2 = \{a \in A \mid a = f(v_1) = f(v_2) \text{ for some } v_1 \neq v_2\}.$$

Now, aside from the edges from $\mathscr{V}(X_{r+1}) \backslash \mathscr{V}(X_r)$, there are elements of $\mathscr{E}(X_{r+1}) \backslash \mathscr{E}(X_r)$ which join two vertices in $\mathscr{V}(X_r)$. These correspond to the subset of A,

$$A' = \{\{w_1, w_2\} \in A \mid \langle w_1, w_2 \rangle \in \mathscr{E}(X_{r+1})\}.$$

An element of $\pi_r(\mathrm{Aut}_e(X_{r+1}))$ must also stabilize A'. However, we have now summarized the condition that $\sigma \in \mathrm{Aut}_e(X_r)$ be in the image of π_r. Namely,

PROPOSITION 2.3. *$\pi_r(\mathrm{Aut}_e(X_{r+1}))$ is precisely the set of those $\sigma \in \mathrm{Aut}_e(X_r)$ which stabilize each of the collections A_1, A_2, A'.*

Proof. We need only now show that, if σ stabilizes A_1, A_2, A', it does indeed extend to an element of $\mathrm{Aut}_e(X_{r+1})$. For such σ, we define the extension as follows. For each "only child" v, $f(v) \in A_1$ implies $\sigma(f(v)) \in A_1$, so map v to the only child of $\sigma(f(v))$. For each pair of twins v, v', $f(v) \in A_2$ implies $\sigma(f(v)) \in A_2$ so map $\{v, v'\}$ to the twin sons of $\sigma(f(v))$ in either order. By construction, this extension stabilizes the set of edges between $\mathscr{V}(X_r)$ and $\mathscr{V}(X_{r+1}) \backslash \mathscr{V}(X_r)$ (note that $f(v)$ and $\sigma(f(v))$ automatically have the same cardinality as subsets of $\mathscr{V}(X_r)$). That it stabilizes the "new" edges between "old" points was implicit, before the extension, in the condition $\sigma(A') = A'$. ∎

Let $A_0 = A \backslash (A_1 \cup A_2)$. In order to isolate the essential problem, we color the set A with six colors to distinguish the six disjoint regions

$$A_0 \cap A', \quad A_1 \cap A', \quad A_2 \cap A', \quad A_0 \backslash A', \quad A_1 \backslash A', \quad A_2 \backslash A'.$$

(The astute reader might observe that only five of these cases can actually occur. That fact is immaterial to the present discussion.) We are now looking for the color preserving elements in $\mathrm{Aut}_e(X_r)$ in its action on A. Thus Trivalent Graph Isomorphism is polynomial-time reducible to the following:

Problem 1. *Input*: A set of generators for a 2-subgroup, G, of $\mathrm{Sym}(A)$, where A is a colored set. *Find*: A set of generators for the subgroup $\{\sigma \in G \mid \sigma \text{ is color preserving}\}$.

2.2. *The Color Automorphism Algorithm for 2-Groups*

The presence of a group action on a set suggests two divide-and-conquer mechanisms: the decomposition of the set into orbits and, in the transitive case, the decomposition of the set into blocks of imprimitivity. Both of these come into play in the algorithm for Problem 1 but they require a generalization of the problem that admits a recursive procedure.

We fix a colored set with n elements. The number and distribution of colors is unimportant. For $a, b \in A$, the relation "a has the same color as b" will be abbreviated "$a \sim b$". Suppose $B \subseteq A$ and $K \subseteq \mathrm{Sym}(A)$.

DEFINITION. Set $\mathscr{C}_B(K) = \{\sigma \in K \mid \text{for all } b \in , \sigma(b) \sim b\}$.

The following properties are immediate

(i) $\mathscr{C}_B(K \cup K') = \mathscr{C}_B(K) \cup \mathscr{C}_B(K')$.

(ii) $\mathscr{C}_{B \cup B'}(K) = \mathscr{C}_{B'}\mathscr{C}_B(K)$.

The generalization we need of Problem 1 is

Problem 2. *Input*: Generators for a 2-subgroup G of $\mathrm{Sym}(A)$, a G-stable subset B, and $\sigma \in \mathrm{Sym}(A)$. *Find*: $\mathscr{C}_B(\sigma G)$.

Problem 1 is the special case $B = A$, $\sigma = 1$. We observe first that

LEMMA 2.4. *If $\mathscr{C}_B(\sigma G)$ is not empty then it is a left coset of the subgroup $\mathscr{C}_B(G)$.*

Proof. The G-stability of B guarantees that $\mathscr{C}_B(G)$ is a subgroup. If $\sigma_0 \in \mathscr{C}_B(\sigma G)$ then, in particular, $\sigma G = \sigma_0 G$. For $\tau \in G$, $b \in B$, we know $\tau(b) \in B$ and so $\sigma_0\tau(b) \sim \tau(b)$. Thus $\sigma_0\tau \in \mathscr{C}_B(\sigma_0 G)$ if and only if $\tau \in \mathscr{C}_B(G)$. That is, $\mathscr{C}_B(\sigma_0 G) = \sigma_0 \mathscr{C}_B(G)$. ∎

By the lemma, we expect the program for Problem 2 to accept, as input, a coset of a group and return an answer of "ϕ" or a coset of a group. The cosets would each be specified by a pair consisting of a representative element and a set of generators for the group.

The algorithm for Problem 2 proceeds as follows: If B is the union of G-stable subsets B', B'' then

$$\mathscr{C}_B(\sigma G) = \mathscr{C}_{B''}\mathscr{C}_{B'}(\sigma G).$$

If not, that is, if G acts transitively on B, we recall Lemmas 1.1, 1.3 and write B as the union of two G-blocks, $B = B' \cup B''$. Note, we do not, this time, attempt to compute $\mathscr{C}_{B'}(\sigma G)$ directly; B' is not G-stable. However, we can find in polynomial (in n) time the subgroup H of G which stabilizes B', B''. Then

$$G = H \cup \tau H$$

and so

$$\begin{aligned}\mathscr{C}_B(\sigma G) &= \mathscr{C}_B(\sigma H) \cup \mathscr{C}_B(\sigma\tau H)\\ &= \mathscr{C}_{B''}\mathscr{C}_{B'}(\sigma H) \cup \mathscr{C}_{B''}\mathscr{C}_{B'}(\sigma\tau H).\end{aligned}$$

It is important to observe that Lemma 2.4 guarantees, when both subanswers $\mathscr{C}_B(\sigma H)$ and $\mathscr{C}_B(\sigma\tau H)$ are non-empty, that they must paste together neatly to a single coset of $\mathscr{C}_B(G)$. In such a case, we would have

$$\mathscr{C}_B(\sigma H) = \rho_1\mathscr{C}_B(H), \qquad \mathscr{C}_B(\sigma\tau H) = \rho_2\mathscr{C}_B(H)$$

and the main answer would be expressed

$$\mathscr{C}_B(\sigma G) = \rho_1\langle\mathscr{C}_B(H), \rho_1^{-1}\rho_2\rangle.$$

(The answer must include the right-hand side since $\mathscr{C}_B(H)$ and $\rho_1^{-1}\rho_2$ are contained in $\mathscr{C}_B(G)$; on the other hand, the right-hand side clearly contains the two subanswers.)

We have shown how, in the intransitive case, the set breaks into disjoint pieces and we solve one problem on each piece. And, in the transitive case, the computation of $\mathscr{C}_B(\sigma G)$ involves four recursive calls to similar problems on sets B', B'' of half the size. It remains only to examine the case $|B| = 1$. But, if $B = \{b\}$ and $GB = B$ then

$$\begin{aligned}\mathscr{C}_B(\sigma G) &= \sigma G \qquad \text{if} \quad \sigma(b) \sim b\\ &= \varnothing \qquad \text{if} \quad \sigma(b) \not\sim b,\end{aligned}$$

so this is resolved in constant time. Standard induction arguments show that the total algorithm requires only polynomial time.

3. The Bounded Valence Case

3.1. *The Groups That Arise*

We now consider graphs of valence $\leqslant t$ where t is, henceforth, fixed. The procedure of subsection 2.1 generalizes, reducing the isomorphism problem to a certain color automorphism problem. The first hurdle is the abstraction of the crucial properties of the groups.

We review the situation: The reduction to determining the kernel and image of

$$\pi_r\colon \mathrm{Aut}_e(X_{r+1}) \to \mathrm{Aut}_e(X_r)$$

remains intact. The set A now consists of all non-empty subsets of $\mathscr{V}(X_r)$ of size $\leqslant t-1$ and then the "father-map"

$$f\colon \mathscr{V}(X_{r+1})\backslash\mathscr{V}(X_r) \to A$$

has the previous meaning. An element $\sigma \in \mathrm{Aut}_e(X_{r+1})$ now lies in $K_r = \mathrm{kernel}\ (\pi_r)$ if

and only if it stabilizes each set of "tuplets," $f^{-1}(a)$, for $a \in A$. The sets $f^{-1}(a)$ form a partition of $f^{-1}(X_{r+1}) \setminus f^{-1}(X_r)$ and K_r is the direct product

$$K_r = \bigtimes_{a \in A} \mathrm{Sym}(f^{-1}(a)).$$

Each of the factors in the direct product can be specified with at most two generators.

We observe next that $\sigma \in \mathrm{Aut}_e(X_r)$ is in the image of π_r if and only if σ stabilizes, for each $0 \leqslant s \leqslant t-1$, the set of fathers of s-tuplets

$$A_s = \{a \in A \mid |f^{-1}(a)| = s\}$$

as well as the set A' of new edges. Color A, accordingly, with $2t$ colors. The problem is once again one of finding the color automorphisms in $G = \mathrm{Aut}_e(X_r)$ acting on A.

Now the fact that the groups in the trivalent case were 2-groups was essential in the algorithm of subsection 2.2 (specifically in the decomposition of the set into two blocks of imprimitivity). The proof of that lay in the observation that the kernels K_r were 2-groups. The nature of the kernels in the present situation motivates the following

DEFINITION. For $k \geqslant 2$, let Γ_k denote the class of groups G such that all the composition factors of G are subgroups of S_k.

Notice, in particular, that the prime factors of $|G|$ for G in Γ_k cannot exceed k.

Remark. We can actually get by with the weaker restriction that the composition factors have bounded order. Essentially, that is the statement in the first announcjement of the present result [15] and it is adequate to establish the claim in the title. Indeed, this would avoid the complication of Lemma 3.2. However, the more precise characterization may be of use in future, careful analyses of the algorithm. This version of Γ_k was proposed by L. Babai.

If $N \lhd G$, the Jordan–Hölder Theorem implies that the collection of composition factors for G is the union of those for N and G/N. Hence

LEMMA 3.1. *If $N \lhd G$, then G is in Γ_k if and only if both N and G/N are in Γ_k.*

In order to show Γ_k is closed with respect to extraction of arbitrary subgroups, we need the following lemma. The result is certainly familiar to specialists but, since we have not found a convenient reference, we include a short proof.

LEMMA 3.2. *The subgroups of S_k are in Γ_k.*

Proof. We must show that, if $N \lhd G \subseteq S_k$ with G/N simple, then G/N is isomorphic to a subgroup of S_k. Let G_i be the subgroup of G which fixes all of the points $\{1, 2, \ldots, i\}$. Form the chain of subgroups generated by the G_i and N, that is,

$$N = G_{n-1}N \subseteq \cdots \subseteq G_2N \subseteq G_1N \subseteq G_0N = G.$$

Note that

$$|G_iN : G_{i+1}N| \leqslant |G_i : G_{i+1}| \leqslant k - i \leqslant k.$$

Since $N \subsetneqq G$, there is a smallest integer j such that $G_{j+1}N \subsetneqq G_jN$. The group $G = G_jN$ acts transitively on the set, C, of cosets of G_jN modulo $G_{j+1}N$ (in fact G_j already does). Since $N \lhd G$, N acts trivally on C. Hence, an action of G/N is induced on C. This action is non-trivial since it is transitive and so, since G/N is simple, it is faithful. ∎

We remark that the above lemma suggests a polynomial-time algorithm for producing an embedding of G/N in S_k.

We can now prove

LEMMA 3.3. *If $G \in \Gamma_k$ then any subgroup of G is in Γ_k.*

Proof. Assume $G \in \Gamma_k$. A composition series for G

$$1 = G^m \lhd \cdots \lhd G^2 \lhd G^1 \lhd G^0 = G$$

yields, for any subgroup H, a series

$$1 = G^m \cap H \lhd \cdots \lhd G^2 \cap H \lhd G^1 \cap H \lhd G^0 \cap H = H$$

(which is not necessarily a composition series). It suffices by Lemma 3.1 to show that each quotient $G^i \cap H/G^{i+1} \cap H$ is in Γ_k. However, that quotient is a subgroup of G^i/G^{i+1} which, by assumption, is a subgroup of S_k. ∎

Finally, to relate this class of groups to the present problem, note that $\mathrm{Sym}(f^{-1}(a)) = S_m$ for some $m \leqslant t - 1$. So, the lemmas yield

$$K_r \in \Gamma_{t-1}$$

and by induction.

PROPOSITION 3.4. *For each r, $\mathrm{Aut}_e(X_r) \in \Gamma_{t-1}$.*

Hence Testing Isomorphism of Graphs of Bounded Valence is polynomial-time reducible to the following Problem 3. Here k is fixed.

Problem 3. *Input*: A set of generators for a subgroup, G, of $\mathrm{Sym}(A)$, where $G \in \Gamma_k$ and A is a colored set. *Find*: A set of generators for the subgroup $\{\sigma \in G \mid \sigma$ is color preserving$\}$.

The algorithm for Problem 3 will follow the divide-and-conquer strategy of subsection 2.2. However, we introduce one additional trick. The next two subsections develop the requisite machinery.

3.2. *Primitive Groups in the Class Γ_k*

The property we require is that such groups have p-subgroups of "small" index. Specifically

PROPOSITION 3.5. *There is a computable constant c ($c = c(k)$) such that: If G is a primitive subgroup of S_n and $G \in \Gamma_k$, then, for some prime p, G has a Sylow p-subgroup of index $\leqslant n^c$.*

As is traditional in studies of primitive groups, the proof will distinguish two cases, according to whether the *socle* is abelian or non-abelian. Recall that the socle of a finite group is the subgroup generated by all the minimal normal subgroups.

We review the structure of the socle of a primitive group. In any finite group, a minimal normal subgroup is necessarily a direct product of isomorphic simple groups [12, proof of Satz I.9.13]. Let N be a fixed minimal normal subgroup of the primitive group G. Suppose G possessed a second minimal normal subgroup, N'. Then N and N' would commute, and since normal subgroups of a primitive group are transitive [26, p. 17], we conclude [26, Sect. 1.4]

(i) N is isomorphic to N'.

(ii) N' is precisely the centralizer of N in G.

In particular, (ii) implies there are no more minimal normal subgroups. Hence the socle of G is either N or $N \times N'$ and, in either case, is a direct product of isomorphic simple groups.

If a primitive group $G \subseteq S_n$ has an abelian socle, it is a classical result [12, Sätze II.3.2, II.3.5] that, for some prime p, $n = p^d$ and G may be identified with a subgroup of $AGL(d, p)$, the d-dimensional affine group over $\mathbb{Z}_p$. ($AGL(d, p)$ is generated by the group $GL(d, p)$ of all non-singular linear transformations of $\mathbb{Z}_p^d$ and the group of all translations, i.e., the additive group of $\mathbb{Z}_p^d$). The translations form the socle of G. One knows also that

$$|AGL(d, p)| = p^{(d(d+1))/2}(p-1)(p^2-1)(p^3-1)\cdots(p^d-1). \qquad (*)$$

(See [12, p. 178], for example, for a discussion of $|GL(d, p)|$.)

We need the following number-theoretic lemma.

LEMMA 3.6. *Let p, q be distinct primes. There is a constant α ($\alpha = \alpha(p, q)$) such that if q^x divides $|AGL(d, p)|$ then $x < d\alpha$.*

Proof. Let $\gamma(y)$ denote the exponent in the highest power of q dividing $p^y - 1$. So

$$p^y = 1 + aq^{\gamma(y)} \qquad \text{with} \qquad (a, q) = 1$$

Then, for any z,

$$\begin{aligned} p^{yz} &= (1 + aq^{\gamma(y)})^z \\ &\equiv 1 + azq^{\gamma(y)} \bmod q^{2\gamma(y)}. \end{aligned}$$

Thus, we derive for $\gamma(y) \geqslant 1$,

(i) If $(z, q) = 1$ then $\gamma(yz) = \gamma(y)$.

(ii) $\gamma(yq) \geqslant \gamma(y) + 1$ and, if $\gamma(y) \geqslant 2$, equality holds.

Now let r be the order of p modulo q and set $s = \gamma(r)$, $t = \gamma(rq)$. Then $s \geqslant 1$ and, by (ii), $t \geqslant s + 1 \geqslant 2$. (Actually, the equality $t = s + 1$ fails only when $q = 2$ and $s = 1$.) We use these relations to describe $\gamma(y)$ in general. Clearly $\gamma(y) = 0$ unless r divides y. By (i) and (ii), if $(u, q) = 1$ then

$$\gamma(ruq^b) = s \qquad \text{if} \quad b = 0$$
$$= t + b - 1 \qquad \text{if} \quad b \geqslant 1.$$

It follows from (*) that the largest x such that q^x divides $|AGL(d, p)|$ is

$$x = s\left[\frac{d}{r}\right] + (t - s)\left[\frac{d}{qr}\right] + \left[\frac{d}{q^2 r}\right] + \left[\frac{d}{q^3 r}\right] + \left[\frac{d}{q^4 r}\right] + \cdots.$$

Erasing the brackets and summing the infinite geometric series, we conclude

$$x < d\left(\frac{s}{r} + \frac{t - s}{qr} + \frac{1}{q(q-1)r}\right). \quad \blacksquare$$

Remark. Actually, much more is known, structurally, about Sylow q-subgroups of $AGL(d, p)$. See [25] for the case $q \neq 2$ and [7] for $q = 2$. The expression for "the largest x" is deducible from those sources.

Then,

PROPOSITION 3.7. *The conclusion of Proposition* 3.5 *holds if the socle of* G *is abelian.*

Proof. We know that $G \subseteq AGL(d, p)$ with $n = p^d$. For each $q \neq p$, the highest power q^x in $|G|$ does not exceed

$$q^{d\alpha(p,q)} = n^{\alpha(p,q)\log_p q}.$$

Hence the product of the powers of all primes $\neq p$ in $|G|$ does not exceed n^c, where

$$c = \operatorname*{Max}_{p \leqslant k}\left(\sum_{\substack{q \neq p \\ q \leqslant k}} \alpha(p, q)\log_p q\right). \quad \blacksquare$$

We turn now to the non-abelian socle case. We shall need the following lemma only when N is the socle of a primitive group. In that instance, it falls out of statements in the appendix of [21].

LEMMA 3.8. *Suppose that the set A admits a faithful, transitive action of a direct product*

$$N = T_1 \times T_2 \times \cdots \times T_r$$

of r nonabelian simple groups. Then $|A| \geqslant 5^r$.

Proof. The result is clear for $r = 0$ or 1 (when $r = 0$, we interpret N as 1). We assume then that $r \geqslant 2$ and that the result holds for groups with fewer than r simple factors. For fixed i, the orbits of T_i form blocks of imprimitivity for N; in particular, they are equal in size. We may assume T_1 has the shortest orbits. Let

$$A = B_1 \cup \cdots \cup B_m$$

be the T_1-orbit decomposition. Denote by K the subgroup of N which stabilizes each of the blocks B_i. Since K is normal in N, it consists of a direct product of some of the T_i's [12, Satz I.9.12]. Without loss of generality

$$K = T_1 \times \cdots \times T_s \qquad \text{for some} \quad s \geqslant 1.$$

Set

$$K' = T_{s+1} \times \cdots \times T_r.$$

Then $K' \cong N/K$ acts faithfully and transitively on the m-element set $\{B_i\}_{1 \leqslant i \leqslant m}$. By the induction hypothesis, $m \geqslant 5^{r-s}$. We consider two cases.

Case 1. $s = 1$. Since T_1 acts faithfully on B_1, $|B_1| \geqslant 5$. Hence $n = |B_1|\, m \geqslant 5 \cdot 5^{r-1} = 5^r$.

Case 2. $s > 1$. Set

$$K'' = T_2 \times \cdots \times T_s.$$

We claim that K'' acts faithfully on B_1. To see this, suppose $\sigma \in K''$ fixes every point in B_1. For each i, $2 \leqslant i \leqslant m$, there is some $\tau \in K'$ with $\tau(B_1) = B_i$. Since σ, τ commute, this implies σ fixes every point in B_i as well. That is, $\sigma = 1$, proving the claim. Since T_1 has the shortest orbits, K'' acts transitively on B_1. Hence T_1, K'' are faithfully represented as commuting transitive subgroups of $\mathrm{Sym}(B_1)$. By [26, Sect. 4], $T_1 \cong K''$ and $|B_1| = |T_1|$. Thus, $s = 2$ and $|B_1| \geqslant 60$. In this case $n = |B_1|\, m \geqslant 60 \cdot 5^{r-2} > 5^r$. ∎

We get a stronger result than Proposition 3.5 in the non-abelian socle case.

PROPOSITION 3.9. *There is a computable constant c $(c = c(k))$ such that: If $G \subseteq S_n$ is primitive with a nonabelian socle and with $G \in \Gamma_k$ then $|G| \leqslant n^c$.*

Proof. The socle is $N = T_1 \times \cdots \times T_r$ where the T_i are isomorphic non-abelian

simple groups. Consider the action of G on N via inner automorphisms and let K be its kernel, that is, the centralizer of N. Since K is normal in G, if it were non-trivial, it would intersect the socle non-trivially. This is not possible since N has trivial center. Hence, G is isomorphic to a subgroup of $\mathrm{Aut}(N)$. The reason for emphasizing this embedding of G is that the structure of $\mathrm{Aut}(N)$ is transparent. The fact that the T_i are the unique minimal normal subgroups of N [12, Satz I.9.12] implies that any automorphism of N consists of a permutation of these factors followed by an automorphism in each factor. In other words, G is a subgroup of

$$\mathrm{Aut}(N) \cong \mathrm{Aut}(T_1) \text{ wr } S_r$$

("wr" indicates wreath product [10, 12]). But since $|G|$ involves only primes $\leqslant k$, the projection of G on S_r has order $\leqslant a^r$, where a is a function of k alone. Thus,

$$|G| \leqslant |\mathrm{Aut}(T_1)|^r a^r.$$

Since $G \in \Gamma_k$, $T_1 \subseteq S_k$, and so $|\mathrm{Aut}(T_1)|$ is bounded. Thus,

$$|G| \leqslant b^r \qquad \text{for} \quad b = b(k).$$

On the other hand, $n \geqslant 5^r$ by Lemma 3.8. Therefore,

$$|G| \leqslant n^{\log_5 b}. \quad \blacksquare$$

The proof of Proposition 3.5 is immediate from Propositions 3.7, 3.9. In the non-abelian socle case we can use any Sylow subgroup, even 1.

3.3. *Finding the p-Subgroup*

The previous section established that, for a certain class of permutation groups, the existence of Sylow p-subgroups of small index is guaranteed. The question remains whether we can find generators for such a subgroup in polynomial time. The problem appears similar to the one mentioned in Lemma 1.2(iii). The difficulty is that the description "P is a Sylow p-subgroup" does not define P uniquely and so it does not yield an effective membership test. However, given (generators for) some p-subgroup P and an element $\sigma \in G$ one can test whether σ and P lie in any common Sylow p-subgroup by seeing whether the order of $\langle \sigma, P \rangle$ is a power of p. This is used for

PROPOSITION 3.10. *Let c be fixed. There is a polynomial-time algorithm for finding generators for a Sylow p-subgroup of $G \subseteq S_n$ provided $|G| = p^s m$ with $m \leqslant n^c$.*

The algorithm: We build, simultaneously, a set Π of generators for a Sylow p-subgroup P and a complete set C of left coset representatives for G mod P. To start

$$\Pi \leftarrow \emptyset, \qquad C \leftarrow \{1\}, \qquad P \leftarrow 1.$$

We use the following subroutine. The input, α is an element of G.

procedure p-Build (α)
  if for some $\gamma \in C$, $\langle \gamma^{-1}\alpha, \Pi \rangle$ is a p-group
    then if $\langle \gamma^{-1}\alpha, \Pi \rangle = P$
      then return
      else add $\gamma^{-1}\alpha$ to Π and let $P = \langle \Pi \rangle$
    else add α to C
  return

Let Φ be the given set of generators for G. We call p-Build(α) for all α in ΦC. Of course, such a call may then result in an increase of C. However, at any point in the construction, $P = \langle \Pi \rangle$ is a p-group and the elements of C are pairwise incongruent modulo any Sylow p-subgroup containing P. In particular, there are never more than m elements in C. Thus the process terminates in polynomial time. When it does halt $\Phi C \subseteq CP$ and so CP is closed under left multiplication by Φ. Hence $CP = G$. Also, since $\langle \gamma, P \rangle$ is not a p-group for any $\gamma \in C$ with $\gamma \neq 1$, P is a Sylow p-subgroup. ∎

Actually, the situation with which we deal in the color automorphism algorithm is covered by

PROPOSITION 3.11. *Let $G \subseteq \mathrm{Sym}(A)$ and suppose $G \in \Gamma_k$. Let B be a G-orbit in A. Then, in polynomial time, we can find a minimal G-block system in B*

$$B = B_1 \cup B_2 \cup \cdots \cup B_m$$

and a subgroup P of G of index $\leqslant m^{c(k)}$ such that P acts on the collection $\{B_1, B_2, \ldots, B_m\}$ as a p-group.

Proof. Since G acts primitively on the blocks, such a P exists by Proposition 3.5. We can find it, for example, by modifying the algorithm in Proposition 3.10 so that it tests whether $\langle \gamma^{-1}\alpha, \Pi \rangle$ acts as p-group on the collection of blocks. ∎

3.4. *The Color Automorphism Algorithm for Groups in Γ_k*

We follow the notation of subsections 2.2, 3.1.
Problem 3 is generalized to

Problem 4. *Input*: Generators for a subgroup G of $\mathrm{Sym}(A)$, where $G \in \Gamma_k$, a G-stable subset B, and $\sigma \in \mathrm{Sym}(A)$. *Find*: $C_B(\sigma G)$.

As before, if B is the union of disjoint G-stable subsets B', B'' then

$$\mathscr{C}_B(\sigma G) = \mathscr{C}_{B''}\mathscr{C}_{B'}(\sigma G).$$

If not, then we find a minimal G-block system in B

$$B = B_1 \cup B_2 \cup \cdots \cup B_m.$$

This time we do not have such a convenient hold on m. Instead, we take advantage of Propositions 3.5, 3.11 and locate a subgroup P with $[G : P] \leqslant m^c$ such that P acts as

a p-group on the collection of blocks $\mathscr{B} = \{B_1, \ldots, B_m\}$. Writing G as a union of $\leqslant m^c$ cosets of P

$$G = \bigcup_i \tau_i P$$

(the τ_i come "for free" in the construction of P) the problem breaks up similarly

$$\mathscr{C}_B(\sigma G) = \bigcup_i \mathscr{C}_B(\sigma \tau_i P).$$

We continue now on each coset of P, except that, to capitalize on Lemma 1.1, we keep the integrity of the individual blocks, B_i, as long as possible. More precisely, the divide-and-conquer is applied to the action of P on $\mathscr{B}$ (a p-group action) not on B. Thus, if $\mathscr{B}$ is a disjoint union of P-stable subcollections $\mathscr{B}', \mathscr{B}''$ we solve the problem sequentially on these. If not, we find a minimal P-block system in $\mathscr{B}$. This time such a system will consist of precisely p subcollections, and the subgroup P' which stabilizes all of the subcollections will have index precisely p. So in the transitive P case, the problem for P on $\mathscr{B}$ breaks into p^2 similar problems for subgroups of P on subcollections of size $|\mathscr{B}|/p$. Each of these is, at worst, equivalent to p^2 similar problems on subcollections of size $|\mathscr{B}|/p^2$, etc. We continue in this fashion until the subcollection consists of precisely one of the original B_i. Having exhausted the p-group action, we are faced, finally, with a problem of the form $\mathscr{C}_{B_i}(\bar{\sigma}\bar{P})$, where $\bar{P}$ is the residual group. However, $|B_i| = |B|/m$. The important observation now is that the problem for each coset of P has been converted to at most m^2 problems on sets of size $|B|/m$. Thus, the original problem for G on B has been converted to at most $m^c \cdot m^2 = m^{c+2}$ problems on sets of size $|B|/m$. The results of subsections 1.2 and 3.3 guarantee that the cost of each reduction is bounded by a polynomial in n, justifying the claim in the title.

We summarize the algorithm for $\mathscr{C}_B(\sigma G)$ below. The special handling of P makes it convenient to describe the computation in two routines, one for $\mathscr{C}_B(\sigma G)$ and one for

$$\mathscr{C}_{\mathscr{B}}(\sigma P) = \mathscr{C}_B(\sigma P),$$

in which P is known to act as p-group on $\mathscr{B} = \{B_1, \ldots, B_m\}$ and $B = \bigcup_i B_i$. The reader will pobably discern that the routines have a common generalization.

I. *Computation of* $\mathscr{C}_B(\sigma G)$

Input: A colored set A, generators for $G \subseteq \mathrm{Sym}(A)$, $\sigma \in \mathrm{Sym}(A)$, a G-stable subset B.

Output: $\mathscr{C}_B(\sigma G)$.

Method:

(i) If $B = \{b\}$,

$$\mathscr{C}_B(\sigma G) = \varnothing \quad \text{if} \quad \sigma(b) \not\sim b$$
$$= \sigma G \quad \text{if} \quad \sigma(b) \sim b.$$

(ii) If B is the union of disjoint G-stable subsets B', B''

$$\mathscr{C}_B(\sigma G) = \mathscr{C}_{B'}\mathscr{C}_{B''}(\sigma G).$$

(iii) If G is transitive on B and $|B| > 1$, find a minimal G-block system in B

$$B = B_1 \cup \cdots \cup B_m.$$

Locate subgroup P such that P acts on $\mathscr{B} = \{B_1, \ldots, B_m\}$ as a p-group and $|G : P| \leqslant m^c$. Decompose G,

$$G = \bigcup_i \tau_i P.$$

Then

$$\mathscr{C}_B(\sigma G) = \bigcup_i \mathscr{C}_{\mathscr{B}}(\sigma \tau_i P).$$

II. *Computation of* $\mathscr{C}_{\mathscr{B}}(\sigma P)$

Input: A colored set A, generators for $P \subseteq \mathrm{Sym}(A)$, $\sigma \in \mathrm{Sym}(A)$, a P-stable collection $\mathscr{B}$ of disjoint subsets of A with P acting as a p-group on $\mathscr{B}$.

Output: $\mathscr{C}_{\mathscr{B}}(\sigma P)$.

Method:

(i) If $\mathscr{B} = \{B_0\}$, $\mathscr{C}_{\mathscr{B}}(\sigma P) = \mathscr{C}_{B_0}(\sigma P)$.

(ii) If $\mathscr{B}$ is the union of disjoint P-stable subcollections $\mathscr{B}'$, $\mathscr{B}''$

$$\mathscr{C}_{\mathscr{B}}(\sigma P) = \mathscr{C}_{\mathscr{B}'}\mathscr{C}_{\mathscr{B}''}(\sigma P).$$

(iii) If P is transitive on $\mathscr{B}$ and $|\mathscr{B}| > 1$, find a minimal p-block system in $\mathscr{B}$

$$\mathscr{B} = \mathscr{B}_1 \cup \cdots \cup \mathscr{B}_p.$$

Locate the subgroup P' which stabilizes all the subcollections $\mathscr{B}_i$. Decompose P,

$$P = \bigcup_{i=1}^{p} \tau_i P'.$$

Then

$$\mathscr{C}_{\mathscr{B}}(\sigma P) = \bigcup_{i=1}^{p} \mathscr{C}_{\mathscr{B}}(\sigma \tau_i P').$$

(The pieces of the right hand side then feed into (ii).)

Remark. We emphasize again that Lemma 2.4 guarantees our ability to combine the answers in I(iii) or II(iii) into a single coset. If several nonempty subanswers are

returned they will all be cosets of the same subgroup and must add up to a coset of a group. That is

$$\bigcup_{i=1}^{s} \rho_i H$$

will be expressible as

$$\rho_1 \langle H, \{\rho_1^{-1}\rho_i\}_{1<i\leqslant s}\rangle.$$

4. Remarks

4.1. *A Faster Trivalent Algorithm*

The trivalent algorithm was presented here in a manner which clearly justifies the polynomial-time claim and which easily generalizes. However, with somewhat more effort, one can make several ad-hoc modifications which improve the efficiency. Our best algorithm for determining whether two connected n-vertex trivalent graphs are isomorphic requires $O(n^5)$ steps [17].

4.2. *Group Intersection*

The general color automorphism problem is polynomial-time reducible to the group intersection problem

Input: Generators for G, $H \subseteq \mathrm{Sym}(A)$.

Find: Generators for $G \cap H$.

The color automorphism problem is the special case when H is the direct product of symmetric groups on the components in a decomposition of A. (It is shown in [16] that, in the general setting, the problems are polynomial-time equivalent.) Thus, the main algorithm of the present paper solves the group intersection problem for these restricted H, and G in Γ_k. However, if G is in Γ_k the restrictions on H can be lifted, still maintaining polynomial time. Outline: Consider the direct product $\mathrm{Sym}(A) \times \mathrm{Sym}(A)$. There are two natural actions of this group on A, employing, respectively, the projections pr_1, pr_2 on the factors, i.e.,

$$\mathrm{pr}_1(\alpha, \beta) = \alpha, \qquad \mathrm{pr}_2(\alpha, \beta) = \beta.$$

This notation is useful in a generalization of the problem which, once again, allows recursion.

Input: Generators of $K \subseteq \mathrm{Sym}(A) \times \mathrm{Sym}(A)$ with $\mathrm{pr}_1(K) \in \Gamma_k$, a $\mathrm{pr}_1(K)$-stable subset B of A, $\sigma \in \mathrm{Sym}(A) \times \mathrm{Sym}(A)$.

Find: $\mathscr{T}_B(\sigma K) = \{\tau \in \sigma K \mid \mathrm{pr}_1(\tau)|_B = \mathrm{pr}_2(\tau)|_B\}$.

In particular,

$$\mathscr{T}_A(G \times H) = \{(\alpha, \alpha) \mid \alpha \in G \cap H\}.$$

Since the nice hypotheses refer only to the pr_1-action, we follow a divide-and-conquer scheme (orbits, blocks, etc.) using that action. The reduction to the base case $|B| = 1$ follows as in Section 3. In that case, we observe that the problem is just one of determining, for some fixed b, c, the elements in the residual K mapping b to c via the pr_2-action. The answer then is either empty or a coset of the stabilizer of the point b.

4.3. *Other Isomorphism Applications*

The basic techniques of Section 3 and Subsection 4.2 apply to a number of graph isomorphism situations. For example,

(i) Colored graphs with bounded color classes, the class considered by Babai [2], come under the scope of the color automorphism algorithm (different "color") for groups in Γ_k. Simply modify the reduction in the introduction so that, instead of S_n, the starting group is the direct product of the symmetric groups of the color classes (a member of Γ_k for k the given bound). Though this, too, puts the problem in P, the algorithm is not as fast as the original.

(ii) Isomorphism of tournaments can be tested in $n^{\log n}$ time [5]. The result involves still another mode of reduction to a color automorphism problem and exploits the fact that tournaments have odd order, therefore solvable, automorphism groups.

(iii) Isomorphism of (v, k, λ)-designs, for bounded λ, can be tested in $n^{\log\log n}$ time. Note that the case $\lambda = 1$ corresponds to projective planes and so this generalizes a result of Miller [19]. The important point to observe about this class is that the number of common neighbors of pairs of points is bounded. This and related results will appear elsewhere.

(iv) We have recently learned that V. Zemlyachenko has used the techniques of this paper to establish an $\exp(n^{1-c})$ upper bound, for some positive c, in general graph isomorphism. See [3] for a discussion of Zemlyachenko's result.

4.4. *New Results on Primitive Permutation Groups*

In our earlier announcement of the main result of this paper [15] we mentioned a new result of Pálfy [20]. He showed that primitive solvable groups have polynomially bounded order. That, and other evidence, prompted our conjecture that primitive groups in Γ_k, k fixed, are also polynomially bounded. We are delighted to learn that this conjecture was confirmed by Babai–Cameron–Pálfy [4]. In fact, they weakened the hypothesis to a bound only on the *nonabelian* composition factors. The significance of all of this is that a simpler version of our main algorithm runs in polynomial time. It is the more obvious generalization of the 2-group situation. When we arrive at a group G acting primitively on m blocks, we can go straight to the subgroup H which stabilizes the blocks and write G as a union of cosets of H. The point is that $[G : H]$ is bounded by a polynomial in m. We observe, nonprejudicially, that the algorithm of Section 3 is faster.

We refer the reader also to [6], in which Cameron draws conclusions about the

orders of primitive groups using the recently completed simple groups classification. In particular, he gives results about the structure of primitive groups in S_n which are larger than $n^{\log n}$; for example, they contain large alternating groups. Thus, there are various hypotheses which can be placed on a class of groups that will avoid the "large" groups and guarantee a subexponential intersection algorithm. It is likely that other complexity applications will be found as well.

4.5. *Recognizing Membership in Γ_k*

Although the result is not needed herein, a natural question arises as to whether membership in Γ_k (k fixed) is polynomial-time decidable (for permutation groups). It is.

To see this, observe that, if G is in Γ_k, then every maximal normal subgroup of G has index $\leqslant b = k!$ With this in mind, we first outline a procedure which, upon input of G, will return a proper normal subgroup N of index $\leqslant b$ or else output "G is not in Γ_k": Take any $b+1$ distinct elements of G, say $\alpha_0,\ldots,\alpha_b$. Then, if G has a normal subgroup of index $\leqslant b$, some two of these elements would be congruent modulo that subgroup. Thus, we generate the normal closures [9], N_{ij}, of $\langle \alpha_i \alpha_1^{-1} \rangle$, for $i \neq j$. If $N_{ij} = G$ for all i, j then G does not have a normal subgroup of index $\leqslant b$, and G is not in Γ_k. Otherwise, we pick up a proper normal subgroup, N'. If $[G : N'] > b$ we continue this process in the following way. Take any $b+1$ elements of G, say, $\alpha_0,\ldots,\alpha_b$, which are pairwise incongruent modulo N'. If G is in Γ_k, N' is contained in a proper normal subgroup of index $\leqslant b$. This time, then, we generate the normal closures of the $\langle \alpha_i \alpha_1^{-1}, N' \rangle$. If none are proper, G is not in Γ_k. Otherwise, we have a larger proper normal subgroup than N', etc.

To test membership of G in Γ_k, we use the above procedure. If N of index $\leqslant b$ is returned, test G/N for membership in Γ_k (in the strict sense of subsection 3.1) by brute force. If G/N is in Γ_k, then it suffices to test N, etc.

There is another polynomial-time procedure for testing membership in Γ_k which is worthy of mention. It is easy to describe but depends upon the Babai–Cameron–Pálfy result noted in 4.4. One reduces the problem to the transitive case (test N, the kernel of the action on an orbit B, and test the image, G/N, of G in $\mathrm{Sym}(B)$) and then to the primitive case (test N, the stabilizer of the blocks, and test the action, G/N, on the set of blocks). If G is in Γ_k it must now be small enough to test by brute force.

Either of these algorithms can be modified to output, for G in Γ_k, a composition series. We wonder whether this could then be utilized in another algorithm for $G \cap H$. For example, assume N is a maximal normal subgroup of G and that we have already found $N \cap H$. We know that $G \cap H / N \cap H$ is a subgroup of G/N. Suppose, in fact, that we knew which subgroup it was (there are only a bounded number of possibilities). Could this be used to recover $G \cap H$?

We mention, finally, that we have a more general algorithm which constructs, in polynomial time, a composition series for an arbitrary permutation group. It requires the simple groups classification, though only for Schreier's conjecture (the outer automorphism group of a finite simple group is solvable). The result will appear elsewhere.

4.6. *Finding Sylow Subgroups*

In subsection 3.3, we dealt with a special situation in which a Sylow p-subgroup could be located in polynomial time. The general problem of finding Sylow p-subgroups is of independent interest and its complexity is open. A related, and perhaps easier, problem is that of finding any element of order p (for a prime p that divides $|G|$). If p is fixed (i.e., bounded), we can view the G-action on A^p; find any instance of some $(a_2,\ldots,a_p,a_1)$ in the orbit of $(a_1,\ldots,a_p)$, and take any σ in G mapping one to the other (an appropriate power of σ has order p exactly). For general p, the best we can do, at this writing, is $n^{\log n}$ time. The algorithm uses, among other things, the general composition-factor algorithm mentioned above. Even then, we do not know a subexponential way of expanding the algorithm to produce Sylow p-subgroups.

4.7. *Certificates*

A *certificate* for a graph is a complete invariant of the isomorphism class, e.g., the min lex adjacency matrix. In most known instances of graph isomorphism algorithms, certificates are attainable about as cheaply [18, 14]. In [2], Babai asked whether his group-theoretic approach could be used to solve this, potentially more useful, question. This has been answered affirmatively for Babai's graphs by the author and P. Klingsberg [13]. However, we do not know whether the techniques of the present paper can be extended to find certificates for graphs of bounded valence. We remark that the relation between the reduction of general graph isomorophism to a color automorphism problem and the reduction of subsections 2.1, 3.1 has an analogue for certificates. A certificate for general graphs could follow from an algorithm for

Input: Generators for $G \subset S_n$, an n-digit binary number m.

Find: The greatest number in the G-orbit of m, where S_n acts on n-digit numbers by permuting digits.

One could, for example, apply such an algorithm to the action of S_n on adjacency matrices, $A \mapsto PAP^t$ (P is a permutation matrix). Tricks like those proposed herein can be utilized to derive certificates for graphs of bounded valence from a solution to the above problem for groups in Γ_k. However, the complexity of the problem is open even for 2-groups.

ACKNOWLEDGMENTS

The author takes pleasure in acknowledging the hospitality of the Department of Computer Science, Cornell University, where he learned much about graph isomorphism, ad other things, in exciting discussions with J. Hopcroft. Thanks are owed also to L. Babai and P. Klingsberg for stimulating conversations and correspondence during the evolution of these ideas.

Note added in proof. We update some of the remarks of Section 4. The $O(n^5)$ bound for trivalent graphs (Section 4.1) was improved to $O(n^4 \log n)$ by C. Schnorr and A. Weber, to $O(n^4)$ by C. Hoffman and, most recently, to $O(n^3 \log n)$ by Z. Galil, E. Luks, C. Schnorr, and A. Weber.

Zemlyachenko's bound for general graph isomorphism (Section 4.3) was $\exp(n^{3/4+o(1)})$. This was

improved by Babai to $\exp(n^{2/3+o(1)})$ and then by the present author to $\exp(n^{1/2+o(1)})$. The latter improvement utilizes a faster ($O(n^{cd\log d})$) algorithm for graphs of valence d.

We have shown that the problem stated in Section 4.7 is *NP*-Hard even if G is an elementary Abelian 2-group.

REFERENCES

1. M. D. ATKINSON, An algorithm for finding the blocks of a permutation group, *Math. Comp.* **29** (1975), 911–913.
2. L. BABAI, Monte-Carlo algorithms in graph isomorphism testing, manuscript, 1979.
3. L. BABAI, Moderately exponential bound for graph isomorphism, *in* "Proceedings Conf. on Fund. Comp. Thy., Szeged (1981)," Lecture Notes in Computer Science, Springer-Verlag, Berlin/New York, in press.
4. L. BABAI, P. J. CAMERON, AND P. P. PÁLFY, On the order of primitive permutation groups with bounded nonabelian composition factors, to appear.
5. L. BABAI AND E. LUKS, A subexponential algorithm for tournament isomorphism, to appear.
6. P. J. CAMERON, Finite permutation groups and finite simple groups, *Bull. London Math. Soc.* **13** (1981), 1–22.
7. R. CARTER AND P. FONG, The Sylow 2-subgroups of the finite classical groups, *J. Algebra* **1** (1964), 139–151.
8. M. FURST, J. HOPCROFT, AND E. LUKS, "A Subexponential Algorithm for Trivalent Graph Isomorphism," Tech. Report 80-426, Computer Science, Cornell Univ., 1980.
9. M. FURST, J. HOPCROFT, AND E. LUKS, Polynomial-time algorithms for permutation groups, *in* "21st IEEE Symp. on Foundations of Comp. Sci. (1980)," pp. 36–41.
10. M. HALL, "The Theory of Groups," Macmillan Co., New York, 1959.
11. C. M. HOFFMAN, Testing isomorphism of cone graphs, *in* "Proc. 12th Symp. Thy. Comp.," pp. 244–251, ACM, New York, 1980.
12. B. HUPPERT, "Endliche Gruppen I," Springer-Verlag, Berlin, 1967.
13. P. KLINGSBERG, E. LUKS, Succinct certificates for a class of graphs, to appear.
14. R. J. LIPTON, The beacon set approach to graph isomorphism, *SIAM J. Comput.* **9** (1980).
15. E. LUKS, Isomorphism of graphs of bounded valence can be tested in polynomial time, *in* "21st IEEE Symp. Found. Comp. Sci. (1980)," pp. 42–49.
16. E. LUKS, The complexity of permutation group problems, to appear.
17. E. LUKS, A faster algorithm for trivalent graph isomorphism, to appear.
18. G. MILLER, Graph isomorphism, general remarks, *J. Comput System Sci.* **18** (1979), 128–142.
19. G. L. MILLER, On the $n^{\log n}$ isomorphism technique, *in* "Proc. 10th Symp. Thy. Comp.," pp. 51–58, ACM, New York, 1978.
20. P. P. PÁLFY, A polynomial bound for the orders of primitive solvable groups, to appear.
21. L. L. SCOTT, Representations in characteristic p, in "The Santa Cruz Conference on Finite Groups," pp. 319–332, Amer. Math. Soc., Providence, R.I., 1980.
22. C. C. SIMS, Graphs and finite permutation groups, *Math. Z.* **95** (1967), 76–86.
23. C. C. SIMS, Computational methods for permutation groups, *in* "Computational Problems in Abstract Algebra," pp. 169–184, Pergamon, Oxford, 1970.
24. C. C. SIMS, "Some Group-Theoretic Algorithms," Lecture Notes in Math. No. 697, Springer-Verlag, Berlin, pp. 108–124, 1978.
25. A. J. WEIR, Sylow p-subgroups of the classical groups over finite fields with characteristic prime to p, *Proc. Amer. Math. Soc.* **6** (1955), 529–533.
26. H. WIELANDT, "Finite Permutation Groups," Academic Press, New York, 1964.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium