SIAM J. ALG. DISC. METH.
Vol. 1, No. 2, June 1980

0196-5212/80/0102-0009 $01.00/0

THE COMPLEXITY OF COLORING CIRCULAR ARCS AND CHORDS*

M. R. GAREY,† D. S. JOHNSON,† G. L. MILLER‡ AND C. H. PAPADIMITRIOU‡

Abstract. The word problem for products of symmetric groups, the circular arc graph coloring problem, and the circle graph coloring problem, as well as several related problems, are proved to be *NP*-complete. For any fixed number *K* of colors, the problem of determining whether a given circular arc graph is *K*-colorable is shown to be solvable in polynomial time.

1. Introduction. The *NP*-completeness of many standard graph-theoretic problems for general graphs [4] has motivated the study of various special classes of graphs for which these problems might be less difficult. A variety of results, both positive (i.e., polynomial time algorithms) and negative (i.e., proofs of *NP*-completeness), have been obtained for such classes as planar graphs, comparability graphs, interval graphs, chordal graphs, circular arc graphs, and circle graphs (see [4]). However, a number of significant questions have remained open. In this paper we address two of these open questions, namely the questions of how difficult it is to *color* circular arc graphs and circle graphs.

A graph G is called a *circular arc graph* if its vertices can be placed in one-to-one correspondence with a family F of arcs of a circle in such a way that two vertices of G are joined by an edge if and only if the corresponding two arcs in F intersect one another. For example, the graph in Fig. 1(a) is a circular arc graph because it has the circular arc model shown in Fig. 1(b). Circular arc graphs were first discussed in [8] as a natural

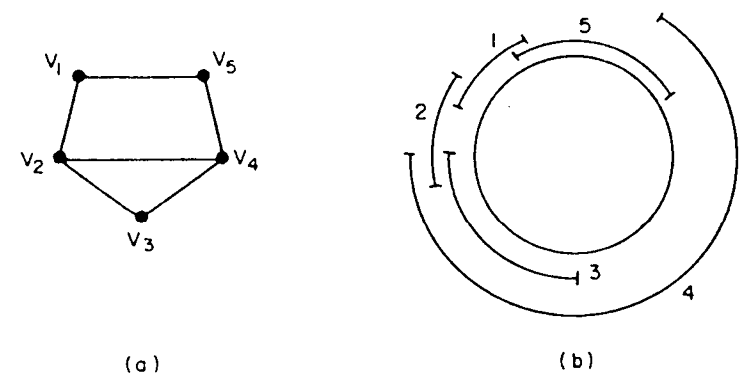

FIG. 1. *A circular arc graph and its circular arc model.*

generalization of *interval graphs* (defined analogously, but using intervals on a line instead of arcs of a circle), and they have since been studied extensively [6], [10], [11], [12], [13]. Tucker [13] has recently given a polynomial time algorithm for recognizing circular arc graphs. Gavril [6] has shown that the problems of finding a maximum independent set, a maximum clique, and a minimum covering by cliques, all of which are *NP*-complete for general graphs, can be solved in polynomial time for circular arc graphs.

* Received by the editors November 19, 1979.
† Bell Laboratories, Murray Hill, New Jersey 07974.
‡ Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

A graph G is called a *circle graph* if its vertices can be placed in one-to-one correspondence with a family of *chords* of a circle in such a way that two vertices are joined by an edge G if and only if the corresponding chords intersect. Fig. 2 shows a circle graph and its chord model. Although no polynomial time recognition algorithm is

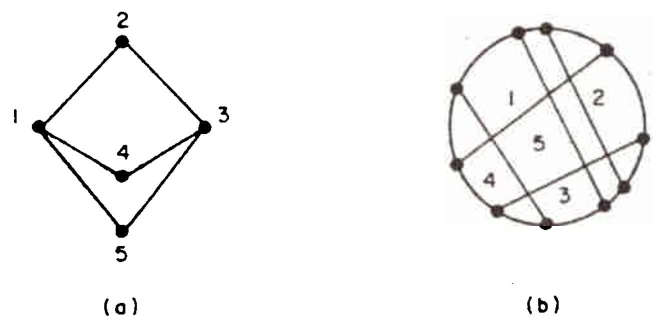

FIG. 2. *A circle graph and its chord model.*

known for circle graphs, [5] shows that, if the graph is described by giving its chord model, then both the maximum independent set problem and the maximum clique problem can be solved in polynomial time.

We shall study the coloring problems for these two classes in the following form: Given a family F of circular arcs (chords) and a positive integer K, can the arcs (chords) in F be colored with K or fewer colors so that no two intersecting arcs (chords) have the same color?

A number of partial results about the arc coloring problem (coloring circular arc graphs) can be found in [12], which also notes the potential applicability of circular arc coloring to the following register allocation problem. Consider a loop in a computer program, and regard the flow of control around the loop as being described by a circle. For each assignment of a value to a variable within the loop, the *lifetime* of that assignment consists of the portion of the loop that begins where the assignment is made and that ends where that value is used for the last time. Each such lifetime thus corresponds to an arc of the circle. Furthermore, a K-coloring of this set of arcs can be regarded as assigning one of K registers to each lifetime, in such a way that, if the value corresponding to that lifetime is stored in the associated register, then no value will ever have to be recomputed or stored elsewhere. The minimum value of K for which the circular arc graph can be colored therefore gives the minimum number of registers needed for doing this.

The chord coloring problem is discussed in [2], where it is shown to model a problem of realizing a given permutation using a minimum number of parallel stacks.

In this paper we provide strong evidence that neither of these coloring problems can be solved in polynomial time, by showing that they are both NP-complete. (Readers unfamiliar with the central notions and terminology pertaining to the theory of NP-completeness can consult [1] or [4].) We begin by concentrating on the circular arc coloring problem. In § 2 we show that this problem is equivalent to the word problem for products of symmetric groups[1] and use this equivalence to derive an

$O(n \cdot K! \cdot K \cdot \log K)$ algorithm for coloring n circular arcs with K colors (whenever possible). This implies that there is a sense in which circular arc coloring is easier than general graph coloring. The algorithm will run in polynomial time for any *fixed* value of K, whereas for general graphs the coloring problem is *NP*-complete for *every* fixed value of $K > 2$ [3]. However, if K is not fixed, the circular arc coloring problem loses its advantage and becomes, like the general problem, *NP*-complete. We prove this in § 3 by showing that the word problem for products of symmetric groups is itself *NP*-complete. In § 4 the *NP*-completeness of the chord coloring problem is derived by a direct transformation from the circular arc coloring problem. Finally, in § 5, we discuss the implications of our results and some of the remaining open problems and directions for further research.

2. Circular arc coloring as a permutation problem. In this section, we formalize the circular arc coloring problem (in a manner suitable for computation), introduce the word problem for products of symmetric groups, and prove that these two problems are equivalent with respect to polynomial time solvability. We then use this equivalence to give an $O(n \cdot K! \cdot K \cdot \log K)$ algorithm for coloring n circular arcs with K colors whenever such a coloring is possible.

We formalize the circular arc coloring problem as follows: A *family* F of *circular arcs* is a set $\{A_1, A_2, \cdots, A_n\}$, where each A_i is an ordered pair (a_i, b_i) of positive integers, with $a_i \neq b_i$. Let m denote the largest integer among all the a_i's and b_i's. Then we can regard the circle as being divided into m parts by m equally spaced points, numbered clockwise as $1, 2, \cdots, m$, and each $A_i = (a_i, b_i)$ can be regarded as representing the circular arc from point a_i to point b_i, again in the clockwise direction. Notice that we might have either $a_i < b_i$ or $b_i < a_i$ for any A_i.

The *span* $sp(A_i)$ of an arc $A_i = (a_i, b_i)$ is the set $\{a_i+1, a_i+2, \cdots, b_i\}$ if $a_i < b_i$ or $\{a_i+1, \cdots, m, 1, 2, \cdots, b_i\}$ if $b_i < a_i$. We say that two arcs A_i and A_j *intersect* if $sp(A_i) \cap sp(A_j)$ is not empty. Notice that two arcs do not intersect if they share only common endpoints. The *circular arc graph* corresponding to the family F is the graph $G = (F, E)$, where $\{A_i, A_j\} \in E$ if and only if A_i and A_j intersect.

Notice that, since we are only interested in the intersection pattern among arcs in F, there is no loss in generality in assuming that all the integers appearing in the pairs (a_i, b_i) are bounded above by $2n$, where n is the numbers of arcs in F. (If not, we can simply sort the a_i's and b_i's and replace each by its rank in the sorted sequence.) Henceforth we shall restrict our attention to families F satisfying this property. The arc coloring problem can now be defined as follows:

ARC COLORING. Given a family F of circular arcs and a positive integer K, can F be partitioned into K classes so that no two arcs in the same class intersect? (Or, equivalently, can the circular arc graph $G = (F, E)$ be colored with K colors?)

To define the word problem for products of symmetric groups, let S_K denote the symmetric group of all permutations on $\{1, 2, \cdots, K\}$ (i.e., the set of all one-to-one functions from $\{1, 2, \cdots, K\}$ onto itself). For $X \subseteq \{1, 2, \cdots, K\}$, let S_X denote the subgroup of S_K consisting of exactly those permutations that leave all elements outside of X fixed. If P_1 and P_2 are subsets of S_K, then their product $P_1 \cdot P_2$ is the set of all permutations $\pi \in S_K$ that can be written as $\pi = \pi_1 \cdot \pi_2$ (with $\pi_1 \cdot \pi_2$ interpreted as first applying π_1 and then applying π_2), where $\pi_1 \in P_1$ and $\pi_2 \in P_2$. The word problem for products of symmetric groups (WPPSG) is defined as follows:

WPPSG. Given K, subsets $X_1, X_2, \cdots, X_m \subseteq \{1, 2, \cdots, K\}$, and a permutation $\pi \in S_K$, does π belong to the set $P = S_{X_1} \cdot S_{X_2} \cdot S_{X_3} \cdots S_{X_m}$, i.e., can π be written as $\pi = \pi_1 \cdot \pi_2 \cdot \pi_3 \cdots \pi_m$ where $\pi_i \in S_{X_i}$ for $1 \leqq i \leqq m$?

The main result of this section is then given by the following theorem:

THEOREM 1. *WPPSG is polynomially equivalent to ARC COLORING.*

Proof. We describe the two transformations. First, given an instance $K, X_1, X_2, \cdots, X_m, \pi$ of WPPSG, we shall show how to construct in polynomial time a family F of circular arcs such that F is K-colorable if and only if $\pi \in P$.

Without loss of generality, we may assume that each integer $i \in \{1, 2, \cdots, K\}$ occurs in at least one set S_j; for if i occurs in no such set, then either $\pi(i) \neq i$ and the answer is trivially "no" for this instance, or $\pi(i) = i$ and we can simply delete i from the instance (decreasing all integers larger than i by 1) to obtain an equivalent instance. The family F will be formed using the points $1, 2, \cdots, K+m$. For each $i \in \{1, 2, \cdots, K\}$, it contains a set F_i of arcs determined by the sets X_j that contain i and a single arc C_i that depends on $\pi^{-1}(i)$. Each F_i is constructed as follows: Let $l_i[1], l_i[2], \cdots, l_i[k(i)]$ denote the indices of the sets X_j that contain i, listed in increasing order. Then F_i consists of the $k(i)$ arcs

$$\begin{aligned} A_{i1} &= (i, K + l_i[1]), \\ A_{i2} &= (K + l_i[1], K + l_i[2]), \\ A_{i3} &= (K + l_i[2], K + l_i[3]), \\ &\vdots \\ A_{i,k(i)} &= (K + l_i[k(i) - 1], K + l_i[k(i)]). \end{aligned}$$

Notice that the spans of the arcs in F_i are pairwise disjoint and that the union of the spans includes exactly the points from $i+1$ up to $K + l_i[k(i)]$. The arc C_i simply spans the region from the end of the last arc in F_i to the beginning of the first arc in $F_{\pi^{-1}(i)}$:

$$C_i = (K + l_i[k(i)], \pi^{-1}(i)).$$

Letting $C = \{C_1, C_2, \cdots, C_K\}$, the family F is defined by

$$F = \bigcup_{i=1}^{K} F_i \cup C.$$

An example of the construction is shown in Fig. 3.

It is easy to see that the family F can be constructed in polynomial time. It remains for us to show that F is K-colorable if and only if $\pi \in P$.

To do this, we first consider all possible ways of K-coloring the alternative family F', which uses the points $1, 2, \cdots, K+m+1$ and which is derived from F by replacing each arc $C_i = (K + l_i[k(i)], \pi^{-1}(i)) \in C$ by the two arcs $(K + l_i[k(i)], K+m+1)$ and $(K+m+1, \pi^{-1}(i))$. Let F'_i, $1 \leqq i \leqq K$, denote the subset of F' that consists of all arcs in F_i, the arc $(K + l_i[k(i)], K+m+1)$, and the arc $(K+m+1, i)$. Then the sets F'_i form a partition of F', and each set F'_i is made up of a collection of pairwise disjoint arcs that together span all the points p, $1 \leqq p \leqq K+m+1$. It follows that at each such point p all K colors must be distributed among the K arcs (one from each F'_i) that span p.

Any K-coloring of F' can be described by a collection of functions σ_p, $1 \leqq p \leqq K+m+1$, where $\sigma_p(j)$ denotes that index $i \in \{1, 2, \cdots, K\}$ such that, among all the arcs spanning point p, color j is assigned to the one from F'_i. Thus each σ_p is a permutation of $\{1, 2, \cdots, K\}$. Without loss of generality we can assume that $\sigma_1(j) = j$ for all j, i.e., that each arc of the form $(K+m+1, i)$ is assigned color i. Furthermore, we observe that in each set F_i' the two arcs $(K+m+1, i)$ and $(i, K+l_i[1])$ both intersect the $K-1$ arcs $(K+m+1, k)$, $i+1 \leqq k \leqq K$, and $(k, K+l_k[1])$, $1 \leqq k \leqq i-1$, so that both these arcs

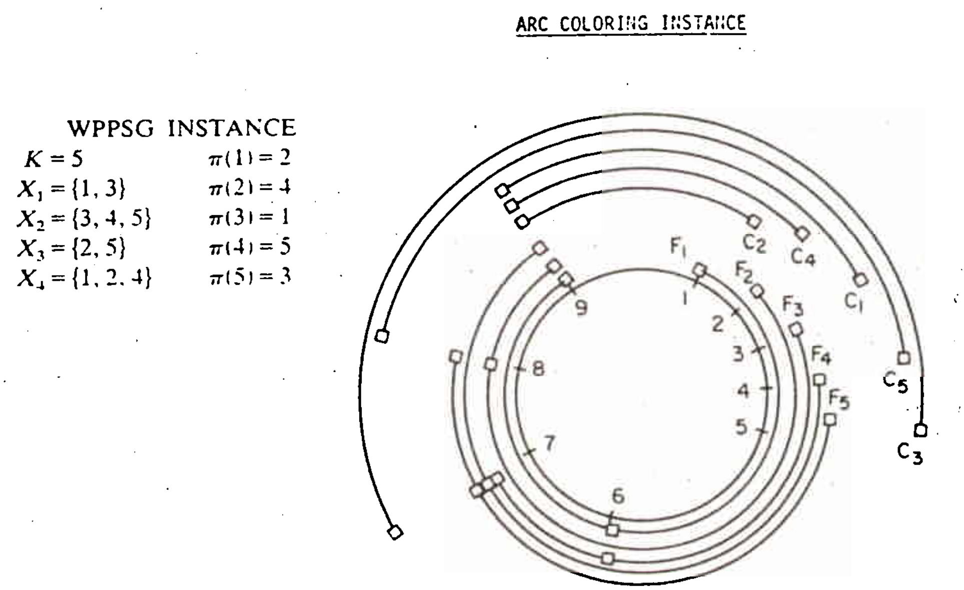

FIG. 3. *An instance of WPPSG and the corresponding instance of ARC COLORING constructed from it.*

must have the same color. Thus, in any K-coloring of F', we have

$$\sigma_1 = \sigma_2 = \cdots = \sigma_{K+1}.$$

We next examine how σ_{p+1} can be formed from σ_p, $K+1 \leqq p < K+m+1$. If $\sigma_p(j) = i$ and the arc from F_i' spanning point p also spans point $p+1$, then we necessarily must have $\sigma_p(j) = \sigma_{p+1}(j) = i$. Thus the only cases in which $\sigma_{p+1}(j)$ can differ from $\sigma_p(j)$ are those in which $F'_{\sigma_p(j)}$ contains an arc that ends (and, by construction, another arc that starts) at the point p. The colors assigned to the sets having this property by σ_p can be arbitrarily redistributed in forming σ_{p+1}. However, by our construction, these are exactly the sets F_i' such that $i \in X_{p-K}$. Therefore, we can write $\sigma_{p+1} = \sigma_p \cdot \pi_{p-K}$ where $\pi_{p-K} \in S_{X_{p-K}}$. Furthermore, any such choice of π_{p-K} provides a legal way of redistributing colors at this point.

Thus the possible "final" permutations σ_{K+m+1} that can be obtained by K-colorings of F' have a particularly simple structure. They are exactly those permutations that can be written as $\pi_1 \cdot \pi_2 \cdot \cdots \cdot \pi_m$ where each π_i belongs to S_{X_i}, i.e., they comprise the set $P = S_{X_1} \cdot S_{X_2} \cdot \cdots \cdot S_{X_m}$.

Recalling that F' was obtained from F by "splitting" each arc $C_i \in C$ into two parts, we observe that the K-colorings of F' that correspond to K-colorings of F are exactly those in which both parts of each C_i are assigned the same color. To interpret this in terms of the σ_p, notice that one "part" of C_i, the arc $(K+l_i[k(i)], K+m+1)$, was placed in the set F_i', whereas the other "part", the arc $(K+m+1, \pi^{-1}(i))$, was placed in the set $F'_{\pi^{-1}(i)}$. Thus, in order for both parts of C_i to have the same color, we must have

$$\sigma_{K+m+1}^{-1}(i) = \sigma_1^{-1}(\pi^{-1}(i)).$$

Since this equality must hold for all $i \in \{1, 2, \cdots, K\}$, and since σ_1 was assumed to be the identity permutation, this implies that a K-coloring of F' corresponds to a K-coloring of F if and only if $\sigma_{K+m+1}^{-1} = \pi^{-1}$, or $\sigma_{K-m+1} = \pi$. But the set of possible values for σ_{K+m+1} is exactly the set P, so F is K-colorable if and only if π belongs to P, which is what we set out to prove.

For the transformation in the other direction, suppose that we are given a family F of circular arcs and a number K of colors. Let m be the largest integer used in the description of the arcs in F. Without loss of generality, we may assume that each point p, $1 \leqq p \leqq m$, is spanned by exactly K arcs from F. If some point p is spanned by more than K arcs, then this can easily be discovered in polynomial time, and it implies that the answer in this instance must be "no." If some point p is spanned by $k < K$ arcs, then we can add $K - k$ arcs of the form $(p-1, p)$ (or $(m, 1)$ if $p = 1$) to F without changing its K-colorability.

Given an F of the above form, we first modify it to form an equivalent family F^* on the points $1, 2, \cdots, K+m$. Let $D_1, D_2, \cdots, D_K$ be any ordering of the arcs in F that span the point 1. Then we replace each arc $(a, b) \in F - \{D_1, D_2, \cdots, D_K\}$ by the arc $(K+a, K+b) \in F^*$, and we replace each arc $D_i = (a, b)$ by the two arcs $(K+a, i)$ and $(i, K+b)$. Since the two arcs replacing each D_i must necessarily have the same color in any K-coloring of F^*, it follows immediately that F^* is K-colorable if and only if F is K-colorable.

The gist of the argument from this point on is that F^* has the same type of structure as the family F constructed in the first half of the proof, so all we need to do is to invert the transformation used there. In order to bring out the structure of F^*, we shall partition it into sets F_i, $1 \leqq i \leqq K$, and C. The set C consists of exactly those arcs in F^* that contain the point 1 in their spans. The sets F_i will be constructed in the order $F_1, F_2, \cdots, F_K$, with a particular F_i being formed by selecting certain arcs from the set

$$R(i) = F^* - C - \bigcup_{j=1}^{i-1} F_j$$

as follows: The first arc selected to be in F_i is the single arc in F^* that has i as its left endpoint. Then, so long as there exists an arc in $R(i)$ whose left endpoint is the same as the right endpoint of the last arc added to F_i, we choose one such arc and add it to $R(i)$. Thus each F_i will consist of a collection of disjoint arcs that span all points from $i+1$ up to some point P_i (and no others). We also index the arcs in C as $C_1, C_2, \cdots, C_K$ in such a way that the left endpoint of arc C_i is the same as the right endpoint P_i of the last arc added to F_i. The fact that every point is spanned by exactly K arcs from F^* enables all of this to be done.

Now we are in a position to construct the sets $X_1, X_2, \cdots, X_m$ and permutation π for the corresponding WPPSG instance. The set X_j consists of those integers $i \in \{1, 2, \cdots, K\}$ such that F_i contains an arc with right endpoint $K+j$. The permutation π has $\pi(i) = j$ if and only if the arc C_j has right endpoint i.

It is not difficult to see that this transformation can be performed in polynomial time. It is also straightforward to verify that if the transformation from the first half of the proof is applied to the WPPSG instance, the resulting ARC COLORING instance is exactly F^*. Hence exactly the same argument as used for that transformation suffices to show that $\pi \in S_{X_1} \cdot S_{X_2} \cdot \cdots \cdot S_{X_m}$ if and only if F^* is K-colorable, and our proof is complete. □

There is an obvious algorithm for solving the WPPSG problem—and, therefore, the ARC COLORING problem, via the transformation of Theorem 1. Given $X_1, X_2, \cdots, X_m$, one simply computes all elements of the set $L = S_{X_1} \cdot S_{X_2} \cdot \cdots \cdot S_{X_m}$ by starting with the set of permutations $P_0 = \{e\}$ and successively constructing $P_{j+1} = \{\pi_1 \cdot \pi_2 : \pi_1 \in P_j \text{ and } \pi_2 \in S_{X_{j+1}}\}$, $j = 0, 1, \cdots, m-1$. The set P is then given by P_m, and we can easily check whether π belongs to it.

To analyze this algorithm, we observe that multiplication of two permutations over $\{1, 2, \cdots, K\}$ can be done in $O(K)$ operations, and, in order to store sets of permutations, we may assume that each permutation $\sigma \in S_K$ is associated with a distinct integer $I(\sigma)$, $1 \leqq I(\sigma) \leqq K!$, in such a way that σ can be computed from $I(\sigma)$ and $I(\sigma)$ computed from σ in $O(K \log K)$ operations (e.g., see [9, pp. 19, 579]). Then the space required for the algorithm is $O(K!)$, and the time required is $O(m \cdot (K!)^2 \cdot K \cdot \log K)$.

This time complexity can be improved, however, by making use of the fact that, if $\pi_1 \in P_j$, $\pi_2 \in S_{X_{j+1}}$, and $\pi_1 \cdot \pi_2 = \pi_1' \in P_j$, then $\pi_1 \cdot S_{X_{j+1}} = \pi_1' \cdot S_{X_{j+1}}$, so we need not compute any of the products involving π_1'. Hence we can compute P_{j+1} from P_j as follows:

Step 1. Select a permutation π_1 from P_j and remove it from P_j.

Step 2. For each permutation $\pi_2 \in S_{X_{j+1}}$,

(a) add $\pi_1 \cdot \pi_2$ to P_{j+1};

(b) remove $\pi_1 \cdot \pi_2$ from P_j (if it's there).

Step 3. If P_j is nonempty, return to Step 1.

This method for computing P_{j+1} from P_j has the property that each product gives us a new member of P_{j+1}. Thus it requires at most $K!$ products and at most $O(K!)$ conversions between a permutation σ and its index $I(\sigma)$. Using this method, the time for the overall algorithm therefore becomes $O(m \cdot K! \cdot K \cdot \log K)$.

The transformation from ARC COLORING to WPPSG given in the proof of Theorem 1 can be implemented easily to run in time $O(K \cdot n)$. Thus we have the following corollary:

COROLLARY. *Deciding whether a family of n circular arcs is K-colorable can be done in* $O(n \cdot K! \cdot K \cdot \log K)$ *time.*

The same time complexity suffices for *constructing* a K-coloring, since in solving the WPPSG instance we can easily save enough information to allow us to reconstruct a sequence of permutations whose product is π. Thus, for any fixed value of K, the circular arc coloring problem can be solved in linear time, and for small values of K the algorithm might actually be practical.

3. ARC COLORING and WPPSG are *NP*-complete. In this section we show that WPPSG is *NP*-complete. By the results of the preceding section, this will imply that ARC COLORING is *NP*-complete. The latter result will in turn imply that CHORD COLORING is *NP*-complete, as we shall see in the next section. In all three cases, we leave to the reader the straightforward verification that the problem in question is in *NP*.

THEOREM 2. *WPPSG is NP-complete.*

Proof. The known *NP*-complete problem that we transform to WPPSG is the following:

DIRECTED DISJOINT CONNECTING PATHS (*DDCP*). Given a directed acyclic graph $G = (V, A)$, an ordering $s_1, s_2, \cdots, s_n$ of the vertices with in-degree 0, and an ordering $t_1, t_2, \cdots, t_n$ of the vertices with out-degree 0 (we may assume that the two sets have the same size), does G contain n mutually vertex-disjoint paths, each going from a distinct s_i to the corresponding t_i, $1 \leqq i \leqq n$?

The undirected version of this problem was proved *NP*-complete by Knuth (see [7]), and the directed acyclic version can be proved *NP*-complete by a trivial modification of his proof.

Suppose we are given an instance $G=(V, A), s_1, s_2, \cdots, s_n$, and $t_1, t_2, \cdots, t_n$ of the DDCP problem. The first step of the transformation is to replace each arc $a=(u, v) \in A$ by two arcs (u, w_a) and (w_a, v), where w_a is a new vertex involved in only these two arcs. This certainly has no effect of the existence of the desired paths. Let $G'=(V', A')$ denote the resulting directed graph.

Now, let $v_1, v_2, \cdots, v_q$ be any topological sorting of the vertices of G', i.e., any ordering such that, for each i, all vertices x for which $(v_i, x) \in A'$ come after v_i in the sequence. Such an ordering can be constructed in time linear in $|A'|$. (See, e.g., [Knuth, Vol. 1]). Furthermore, without loss of generality, we may assume that $v_i = s_i$ for $1 \leqq i \leqq n$ and that $v_{q-n+i} = t_i$ for $1 \leqq i \leqq n$.

For each vertex v_i, let

$$B(i)=\{j:(v_j, v_i) \in A'\}.$$

The sets X_j, $1 \leqq j \leqq m = q-n+1$, for the corresponding WPPSG instance are then defined as follows:

$$X_j=\{n+j\} \cup B(n+j), \qquad 1 \leqq j \leqq q-n$$

$$X_{q-n+1}=\{1, 2, \cdots, q-n\}.$$

The permutation π is defined by:

$$\pi(i)=q-n+i, \qquad 1 \leqq i \leqq n$$

$$\pi(i)=i-n, \qquad n+1 \leqq i \leqq q.$$

This transformation is easily performed in polynomial time. It remains for us to show that $\pi \in P = S_{X_1} \cdot S_{X_2} \cdot \cdots \cdot S_{X_m}$ if and only if the desired paths from each s_i to each t_i exist in G'.

First, let us examine how the WPPSG instance can be interpreted in terms of the graph G'. Each position in a permutation corresponds to a vertex of G'. Initially, each such position/vertex is labeled by its own index. When we apply a permutation π_i from some S_{X_i} we move the labels around on some subset of vertices, specifically on some subset of the vertices whose indices belong to the set X_i. Furthermore, the set X_i contains precisely the indices of vertex v_{n+i} and its immediate predecessors in G'. Thus the process of choosing a sequence of permutations $\pi_1, \pi_2, \cdots, \pi_m$, each $\pi_i \in S_{X_i}$, corresponds exactly to choosing a sequence of label rearrangements, first among v_{n+1} and its immediate predecessors, then among v_{n+2} and its immediate predecessors, and so on, until finally we are allowed to rearrange the labels on all vertices in $V' - \{t_1, t_2, \cdots, t_n\}$. Our goal is to move each label i, $1 \leqq i \leqq n$, all the way from vertex $s_i = v_i$ to the corresponding vertex $t_i = v_{q-n+i}$. Once this has been done, the final permutation can be chosen to arbitrarily rearrange the labels on the vertices outside of $\{t_1, t_2, \cdots, t_n\}$. (In essence, we don't really care what labels end up on these vertices, but the WPPSG problem requires that the entire permutation (i.e., the complete final labeling) be specified.) Thus the permutation π belongs to P if and only if the above relabeling process can be performed in such a way that the label i ends up on vertex $v_{\pi(i)}$, $1 \leqq i \leqq q$.

Given this interpretation, it is not difficult to see that the transformation works as required. Suppose that G' does contain a set of vertex-disjoint paths, one from each s_i to the corresponding t_i, $1 \leqq i \leqq n$. Let $A^* \subset A'$ denote the set of all arcs that occur in these n paths. Notice that, since the paths are disjoint, no vertex will appear more than once as right endpoint of an arc in A^*. The jth step of the corresponding relabeling process, $1 \leqq j \leqq q-n$, is performed as follows: At the jth step we are allowed to rearrange the labels that occur on vertex v_{n+j} and its immediate predecessors. If there is some arc of the form $(u, v_{n+j}) \in A^*$, then we simply interchange the labels on u and v_{n+j}, leaving all other labels where they were. If there is no such arc in A^*, then no labels at all are moved. Since the given paths are disjoint and since V' is indexed in topological order, it is straightforward to verify that this process will succeed in moving each label i, $1 \leqq i \leqq n$, from s_i to t_i by the end of step $q-n$. Step $q-n+1$ then can rearrange the labels on the vertices in $V' - \{t_1, t_2, \cdots, t_n\}$ from where they have been left by the preceding steps to where they are required to be. Thus the existence of the specified disjoint paths implies the existence of the required relabeling sequence, which in turn implies that $\pi \in P = S_{X_1} \cdot S_{X_2} \cdot \cdots \cdot S_{X_m}$.

For the other direction, suppose there exist $\pi_i \in S_{X_i}$, $1 \leqq i \leqq m$, such that $\pi = \pi_1 \cdot \pi_2 \cdot \cdots \cdot \pi_m$, and consider the corresponding relabeling process on G'. For $1 \leqq i \leqq n$, we know that label i starts out on s_i and ends up on the corresponding t_i. What we need to show is that each label i moves only along a path in G' and that the paths for two such labels never intersect at a vertex. For the first of these, suppose that at the jth relabeling step label i is moved, but not along an arc of G' (or not in the proper direction). The topological ordering of V' insures that X_j is the first set to contain v_{n+j}, so label i could not have appeared on v_{n+j} at the beginning of this step. Thus step j must move label i from one immediate predecessor of v_{n+j} to another such immediate predecessor. This implies that v_{n+j} must be one of the original vertices of G, because each of the vertices added to G in forming G' has only *one* immediate predecessor. In this case, however, we know that each immediate predecessor of v_{n+j} has only one arc leaving it in G', the one to v_{n+j}, so no immediate predecessors of v_{n+j} occur in any sets after X_j. Thus such a "parallel move" of label i would prevent it from ever reaching t_i, a contradiction which proves that the labels $1, 2, \cdots, n$ move only along paths in G'. To see that two such paths cannot intersect, we simply need to observe that the only time a label i, $1 \leqq i \leqq n$, can move to a vertex v_{n+j} by moving along an arc of G' is at step j, and only one such label can be moved to v_{n+j} during that step. Thus the paths followed by these labels must be disjoint, and the proof is complete. □

As a consequence of Theorems 1 and 2, we immediately have the following:

COROLLARY. *ARC COLORING is NP-complete.*

We can also make a remark about an interesting special case of WPPSG, that in which each set X_i contains only two elements. This is simply the problem of determining, given a permutation π and a sequence of pairwise interchanges, whether π can be realized by performing some subsequence of the given interchanges. Let us call this problem WPPSG2. We can transform any instance of WPPSG to an equivalent instance of WPPSG2 by replacing each set $X_i = \{a_1, a_2, \cdots, a_l\}$ by the following sequence of $\binom{l}{2}$ two-sets:

$$\begin{array}{l} \{a_1, a_2\}, \{a_1, a_3\}, \quad \cdots \quad, \{a_1, a_l\}, \\ \qquad\qquad \{a_2, a_3\}, \quad \cdots \quad, \{a_2, a_l\}, \\ \qquad\qquad\qquad \vdots \\ \qquad\qquad \{a_{l-2}, a_{l-1}\}, \{a_{l-2}, a_l\}, \\ \qquad\qquad\qquad\qquad \{a_{l-1}, a_l\} \end{array}$$

Thus we have as a corollary of Theorem 2:

COROLLARY. *WPPSG2 is NP-complete.*

4. Chord coloring. We model the chord coloring problem as follows: A *chord*, like a circular arc, is a pair (a, b) of integers. The difference between an arc and a chord lies in the way we interpret such a pair. If the integers occurring in a family of chords (or arcs) are arranged in clockwise order around a circle, the chord (a, b) is viewed as the straight line connecting a and b, whereas the arc (a, b) is viewed as the arc of the circle (in the clockwise direction) from a to b. Note that, in this interpretation, the chords (a, b) and (b, a) are identical, but the arcs (a, b) and (b, a) are different (and complements of each other). We can, however, identify the chord (a, b) with the *shorter* of the two arcs (a, b) and (b, a), i.e., the one with smaller span (as defined in § 2), breaking ties arbitrarily. Then it is easy to see that two chords intersect if and only if the corresponding two arcs A_i and A_j *overlap*, in the sense that their spans intersect and, in addition, neither $sp(A_i) \subseteq sp(A_j)$ nor $sp(A_j) \subseteq sp(A_i)$. In order to avoid any confusion when we use this identification in what follows, we shall always refer to chords as "overlapping" rather than intersecting.

The *circle graph* corresponding to a family $F = \{A_1, A_2, \cdots, A_n\}$ of chords is the graph $G = (F, E)$ where $\{A_i, A_j\} \in E$ if and only if the chords A_i and A_j overlap. The chord coloring problem is then defined as follows:

CHORD COLORING. Given a family F of chords and a positive integer K, can F be partitioned into K classes so that no two chords in the same class overlap? (Or, equivalently, can the circle graph $G = (F, E)$ be colored with K colors?)

The main result of this section shows that CHORD COLORING is at least as hard as ARC COLORING.

THEOREM 3. *CHORD COLORING is NP-complete.*

Proof. We derive this result by showing that ARC COLORING is polynomially transformable to CHORD COLORING. Given an instance, F, K of ARC COLORING, we shall show how to construct in polynomial time a family F' of chords such that the chords in F' are K-colorable if and only if the arcs in F are K-colorable. The idea behind the construction is quite simple and can be summarized as follows: If we view chords in terms of their corresponding arcs, arc coloring and chord coloring are almost identical problems, differing only in cases where one arc is contained in another (see Fig. 4(a)). We are going to *remove* all such occurrences of containment from F by replacing each arc by a sequence of small chords (Fig. 4(b)). However, we must ensure that all small chords replacing a particular original arc behave like a single arc, in the sense that they all must be given the same color. We do this by adding a "clique" of $K-1$ chords at each of the junction points (Fig. 4(c) shows the details around the junction points circled in Fig. 4(b)).

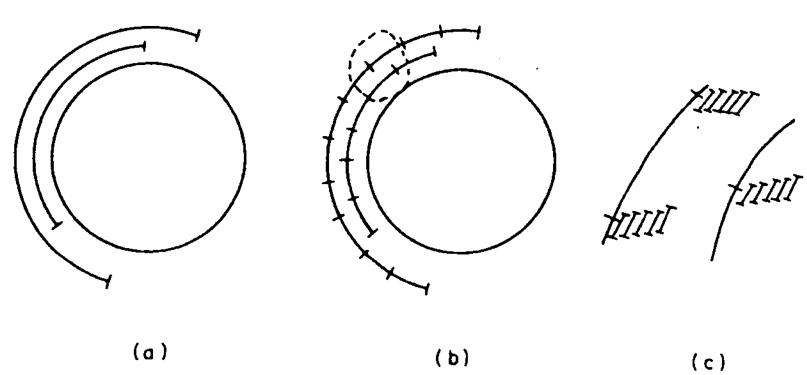

FIG. 4. *An instance of arc containment (a), the result of replacing each arc by a sequence of small "chords" (b), and a "closeup" showing how "cliques" are added at junction points (c).*

Formally, let F be the given family of n arcs, with $m \leqq 2n$ denoting the largest integer used in their descriptions, and let K be the specified number of colors. For each arc $A_i = (a_i, b_i) \in F$ and each point $p \in sp(A_i)$, F' contains the chords

$$(2K(2pn-(n+i)), 2K(2(p+1)n-i)), \qquad p = a_i + 1,$$

$$(2K(2pn-i), 2K(2(p+1)n-i)), \qquad a_i + 1 < p < b_i.$$

Furthermore, it is easy to verify that two original arcs intersect if and only if there are two chords derived from them that overlap. Now consider each pair of chords, derived from the same original arc, that share a common endpoint. By the construction, that common endpoint has the form $2Kx$ for some integer x. We then add the following "clique" chords, all containing the point $2Kx$ in their spans:

$$\begin{gathered}(2Kx-1, 2Kx+K-1),\\ (2Kx-2, 2Kx+K-2),\\ \vdots\\ (2Kx-(K-1), 2Kx+1).\end{gathered}$$

Observe that these $K-1$ "clique" chords all overlap one another and, in addition, they all overlap the two chords that share endpoint $2Kx$. Furthermore, these are the *only* chords that they overlap.

Since each A_i satisfies $|sp(A_i)| \leqq m \leqq 2n$, the above construction clearly can be performed in polynomial time. By sorting all the chord endpoints and replacing each endpoint by its rank in the sorted order, all of which can be done in polynomial time, we also can convert the set of chords into one having the same intersection pattern and having a description using no integer larger than twice the total number of chords. For convenience, however, we shall continue to work with the "un-condensed" version in the remainder of the proof.

We claim that the arcs of F are K-colorable if and only if the chords of F' (using "overlap" instead of "intersect") are K-colorable. Given any K-coloring of F, let $C(A_i)$ denote the color used for arc A_i. Then, for each A_i, we color all the chords in F' derived from A_i with color $C(A_i)$. Since two chords derived from the same arc A_i do not overlap and since two chords derived from different arcs A_i and A_j do not overlap unless A_i and A_j intersect (in which case we know that $C(A_i) \neq C(A_j)$), this "partial" coloring correctly assigns different colors to overlapping chords. All that remains is to color the various "clique" chords. Consider the clique chords surrounding some point $2Kx$ that is a common endpoint of two chords derived from a particular arc A_i. Since these $K-1$ clique chords overlap only one another and two chords already colored with color $C(A_i)$, we may color each of them with a different one of the remaining $K-1$ colors. Doing this for each such set of clique chords, we finally obtain a K-coloring for the chords in F'.

On the other hand, suppose that we have a K-coloring for the chords in F'. Consider any two chords that share a common endpoint and that are derived from the same original arc A_i. These two chords must be assigned the same color, since both overlap all the $K-1$ clique chords surrounding their common endpoint and $K-1$ distinct colors must be used on those clique chords. It follows that, for each original arc A_i, *all* chords in F' derived from A' must have the same color. Thus, we can obtain a K-coloring for the arcs in F by assigning to each arc A_i the same color that is assigned to all the chords derived from A_i. This is a legal K-coloring, because two arcs A_i and A_j in

F intersect only if two chords in F' derived from them overlap, and because the given coloring for F' assigned different colors to any two chords that overlap. □

5. Conclusion. In this paper we have shown that the word problem for products of symmetric groups is *NP*-complete, and from this have derived the *NP*-completeness of graph coloring, even when restricted to circular arc graphs or circle graphs. Although we have not given formal definitions for the register allocation problem and the problem of realizing a permutation with parallel stacks, which were claimed to be equivalent to circular arc graph and circle graph coloring in § 1, the *NP*-completeness of these problems also follows from our results. (The reader may fill in the details by looking up the formal definitions in [2], [12].)

A number of open questions remain. In § 2 we were able to present an algorithm which, for any fixed K, ran in polynomial time and produced a K-coloring of a family of circular arcs if one existed. Does a similar algorithm exist for the chord coloring problem, or is there, as with general graph coloring, some fixed K for which the chord coloring problem is *NP*-complete? What is the complexity of the coloring problem for *proper* circular arc graphs (graphs representable by families of arcs which intersect if and only if they overlap)?

More basically, is there a polynomial time algorithm for recognizing circle graphs and constructing their representations in terms of chords (or arcs)? Such algorithms have been found by Tucker for circular arc graphs [13] and proper circular arc graphs [10]. A similar algorithm for circle graphs might well widen the usefulness of the algorithms in [5], as these assume that the representation of the circle graph is known.

REFERENCES

[1] A. V. AHO, J. E. HOPCROFT AND J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] S. EVEN AND A. ITAI, *Queues, stacks, and graphs*, Theory of Machines and Computations, Z. Kohavi and A. Paz, eds., Academic Press, New York, 1971, pp. 71–86.

[3] M. R. GAREY, D. S. JOHNSON AND L. J. STOCKMEYER, *Some simplified NP-complete graph problems*, Theor. Comput. Sci., 1 (1976), pp. 237–267.

[4] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[5] F. GAVRIL, *Algorithms for a maximum clique and a maximum independent set of a circle graph*, Networks, 3 (1973), pp. 261–273.

[6] ———, *Algorithms on circular-arc graphs*, Networks, 4 (1974), pp. 357–369.

[7] R. M. KARP, *On the complexity of combinatorial problems*, Networks, 5 (1975), pp. 45–68.

[8] V. KLEE, *What are the intersection graphs of arcs in a circle?*, Amer. Math. Monthly, 76 (1969), pp. 810–813.

[9] D. E. KNUTH, *The Art of Computer Programming*, Vol. 3, Addison-Wesley, Reading, MA, 1973.

[10] A. TUCKER, *Matrix characterizations of circular-arc graphs*, Pacific J. Math., 39 (1971), pp. 535–545.

[11] ———, *Structure theorems for some circular-arc graphs*, Discrete Math. 7 (1974), pp. 167–195.

[12] ———, *Coloring a family of circular arcs*, SIAM J. Appl. Math., 29 (1975), pp. 493–502.

[13] ———, *An efficient test for circular-arc graphs*, SIAM J. Comput., 9 (1980), pp. 1–24.