JOURNAL OF COMPUTER AND SYSTEM SCIENCES 29, 225–242 (1984)

On the Computational Complexity of Path Cover Problems

SIMEON NTAFOS* AND TEOFILO GONZALEZ

Computer Science Program, The University of Texas at Dallas, Richardson, Texas 75080

Received October 4, 1982; revised May 1, 1984

In this paper the complexity of a number of path cover problems in acyclic digraphs, acyclic structured digraphs, and rooted trees is considered. The problems deal with finding path covers of a certain size for (a) some elements of the digraph (e.g., required pairs, vertex subsets) or (b) the vertices of the digraph when restrictions are placed on the members of the path cover (e.g., impossible pairs, length-constrained covers). An attempt is made to characterize the complexity of path cover problems in general and a connection between the complexity of a path cover problem and the existence of a reachability relation on the elements that are to be covered in the digraph is pointed out. © 1984 Academic Press, Inc.

I. INTRODUCTION

A path cover in a digraph is a set of paths that covers certain features of the digraph. One important application of path covers is in the area of program testing. Two commonly used testing strategies are segment and branch testing, in which we guarantee that all statements or branches in a program are executed, respectively. If we model the program as a digraph, these strategies correspond to finding path covers for the vertices or edges of the digraph, respectively. In many cases we want a more extensive test set in a program. For example, we may want to test certain interactions between program statements. This situation can be modelled by introducing the notion of required pairs, i.e., $[v_i, v_j]$ is a required pair if we want some test path to visit both statements i and j in the program [1]. Then, testing corresponds to finding a path cover for the required pairs in the digraph. A shortcoming of the digraph model is that paths in the digraph do not always correspond to execution sequences in the program. This gives rise to a second class of path cover problems in which restrictions are placed on the paths that can be in the path cover. We will consider a number of path cover problems with applications in program testing as well as some problems of theoretical interest in trying to characterize the complexity of path cover problems in general.

Let $G = (V, E)$ be an acyclic digraph. We will assume that G has a unique source vertex s and a unique sink vertex t such that there is a path from s to any vertex in G and from any vertex in G to t. A path cover for the vertices of G is a set of s–t (source-to-sink) paths $P = \{p_1, p_2, ..., p_m\}$ such that for each $v_i \in V$ there exists at

* This work was partially supported by the NSF under Grant MCS-8003322.

0022-0000/84 $3.00

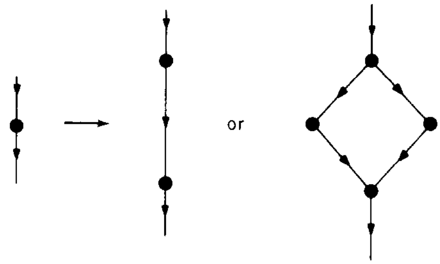

FIG. 1. Acyclic structured digraphs.

least one path $p_j \in P$ that visits v_i. P is a minimum path cover if there exists no path cover P' such that $|P'| < |P|$, where $|X|$ is the cardinality of set X. A vertex v_i reaches a vertex v_j if there exists an s–t path that visits v_i and then v_j. Vertex v_i dominates vertex v_j if every s–t path that visits v_j also visits v_i. Vertices v_i, v_j are said to be incomparable if neither one reaches the other. Dilworth [2] showed that the cardinalities of a minimum path cover for the vertices of G and a maximum set of mutually incomparable vertices are equal. A similar result holds for path covers for the edges of G. Minimum path covers for V can be found efficiently using maximum matching and minimum flow techniques [1; 3].

In the next section we will consider a number of path cover problems in acyclic digraphs and study their complexity. In general, there are two types of problems. In the first, we are looking for a path cover for certain features of the digraph (e.g., required pairs, vertex subsets); in the other, restrictions are placed on the paths that can be included in the path cover and we are looking for a legal (where the meaning of "legal" depends on the specific case) path cover for the vertices of G. We consider these problems for three progressively more restricted types of digraphs: acyclic digraphs, acyclic structured digraphs, and rooted trees. Structured digraphs are introduced to model the programming methodology known as "structured programming." For the purposes of this paper, we define an acyclic structured digraph to be any digraph that can be constructed by sequencing and nesting of the basic alternation control structure (IF-THEN-ELSE) shown in Fig. 1.

In the third section we consider the general problem of finding path covers in an acyclic digraph. It is shown that the complexity of a path cover problem is related to whether or not a reachability relation for the elements that are to be covered can be defined in the digraph. We also attempt to formulate Dilworth-type theorems for each problem.

II. PATH COVER PROBLEMS

1. *Required Pairs and Paths*

Let G be an acyclic digraph with a single source and a single sink. A required pair in G is a pair of vertices $[v_i, v_j]$ such that v_i reaches v_j. Let the set of required pairs of G be: $R=\{r_1, r_2, ..., r_k\} \subseteq \{[v_i, v_j] \mid v_i, v_j \in V, v_i \text{ reaches } v_j\}$. The required pair problem may be stated as follows:

REQPR.
INSTANCE: Acyclic digraph G, set of required pairs R in G, positive integer $m \leqslant |R|$.
QUESTION: Is there a path cover of size m for the required pairs in R?

In [1] it was shown that REQPR belongs to the class of *NP*-complete[1] problems [4]. In fact, REQPR remains *NP*-complete even if G is an acyclic structured digraph [5]. If G is a rooted tree, in a required pair $[v_i, v_j]$, v_i always dominates v_j and there is a unique path that covers v_j and $[v_i, v_j]$. Thus, REQPR reduces to the problem of finding a minimum path cover for a set of vertices in a rooted tree which can be easily solved.

Consider next the required path problem [1]. A required path is a path q_i in G that we want covered. An s–t path p_j is said to cover q_i if q_i is a subpath of p_j. Let the set of required paths of G be $R_{th}=\{q_1, q_2, ..., q_k\}$. Then we have

REQPR: Find a minimum set of s–t paths $P=\{p_1, p_2, ..., p_m\}$ such that for all $q_i \in R_{th}$ there is a path $p_j \in P$ and q_i is a subpath of p_j.

Without loss of generality we may assume that no required path is a subpath of another required path. Let $q_i = v_{i1}(v_{i1},) \cdots (, v_{it})\, v_{it}$ and $q_j = v_{j1}(v_{j1},) \cdots (, v_{jt})\, v_{jt}$ be required paths. As shown in [1], we can define the following reachability relation for the required paths: Required path q_i reaches q_j if

(1) v_{it} reaches v_{j1} or $v_{it}=v_{j1}$, or

(2) $q_i = q_{i1}q_{i2}$, $q_j = q_{j1}q_{j2}$, and $q_{i2}=q_{j1}$.

Then we can solve REQPTH by constructing the acyclic digraph $G_r=(V_r, E_r)$, where $V_r=\{v_1, v_2, ..., v_k\}$ and $(v_i, v_j) \in E_r$ if q_i reaches q_j in G. Note that for each required path in G there is a vertex in G_r and for each path in G_r there is a corresponding path in G. Then, REQPTH is equivalent to finding a minimum path cover for the vertices of an acyclic digraph and can be solved in polynomial time.

2. *Impossible and Must Pairs/Paths*

A second class of minimum path cover problems arises when restrictions are placed on the members of the path cover. For example, if we model a computer program as a digraph, it is not uncommon that some paths in the digraph do not correspond to executable sequences in the program. To model this situation, Krause *et al.* [6] introduced the notion of impossible pairs. A vertex pair $[v_i, v_j]$ is an impossible pair if, because of external considerations, no s–t path is allowed to visit both v_i and v_j although v_i reaches v_j. Similarly we can define impossible paths, must

[1] A problem A is *NP*-complete ($A \in NPC$) if: (a) A can be solved by a nondeterministic Turing machine in polynomial time ($A \in NP$) and (b) some known *NP*-complete problem B can be polynomially transfored to A.

pairs, and must paths [1]. An impossible path in G is a path $p(v_i, v_j)$ that is not allowed to be a subpath of any s–t path. A must pair $[v_i, v_j]$ is a vertex pair such that if an s–t path visits v_i, it must also visit v_j. We will discuss must paths later in this section. Gabow, Maheshwari, and Osterweil [7] showed that finding a legal s–t path in the presence of impossible pairs is NP-complete even if G is an acyclic structured digraph. Also, finding a legal path in the presence of must pairs is NP-complete [1; 5].

Let I_{pr}, I_{pth}, and M_{pr} be sets of impossible pairs, impossible paths, and must pairs specified for an acyclic digraph G. Then we have the corresponding path cover problems IPR, IPTH, and MPR, dealing with determining whether or not there exists a legal path cover of certain size for the vertices of G. Note that the path cover problems IPR and MPR are distinct from the corresponding legal path problems since the existence of a legal s–t path does not imply the existence of a legal path cover in a digraph. In practice, one would expect that finding a legal path in a program is relatively easy. In the following discussion we will assume that a legal path cover exists in each case.

We now show that IPR and MPR are NP-complete even if G is an acyclic structured digraph by transforming the 3-COLORABILITY problem to each of them.

> **3-COLORABILITY** [4].
> **INSTANCE**: Graph $G' = (V', E')$.
> **QUESTION**: Is G' 3-colorable, i.e., does there exist a function $f\colon V' \to \{1, 2, 3\}$ such that $f(v_i) \neq f(v_j)$ whenever $\{v_i, v_j\} \in E'$?

Given an instance of 3-COLORABILITY we construct the acyclic structured digraph $G = (V, E)$ shown in Fig. 2. To each vertex v_i in G' corresponds a subgraph G_i in G that contains vertices v_i and v_i'. The digraph G has $3|V'| + 1$ vertices and thus it can be constructed in polynomial time. Then we have

THEOREM 1. *IPR $\in$ NPC even if G is an acyclic structured digraph.*

Proof. Clearly IPR $\in NP$. Given an instance of 3-COLORABILITY we specify the following set of impossible pairs in the digraph G: $I_{pr} = \{[v_i, v_j] \mid (v_i, v_j) \in E'\}$. Then G' is 3-colorable if, and only if, there exists a legal path cover of size 3 for the vertices of G. Given a 3-coloring in G' we construct a legal path cover $P = \{p_1, p_2, p_3\}$ for V by letting path p_j visit either vertex v_i in G if v_i is assigned color j in G', or vertex v_i' in G if v_i is not assigned color j in G' for $1 \leqslant i \leqslant |V'|$. Conversely, given a path cover $P = \{p_1, p_2, p_3\}$ for V we obtain a 3-coloring in G' by assigning color j to all vertices $v_i \in V'$ such that $v_i \in p_j$ in G and $v_i \notin p_a$ for any $a < j$. Clearly, since a legal path can not visit two vertices $v_i, v_k \in V$ if $\{v_i, v_k\} \in E'$ the resulting coloring is a proper coloring.

A similar result holds for must pairs in acyclic structured digraphs.

THEOREM 2. *MPR $\in$ NPC even if G is an acyclic structured digraph.*

Proof. Again we use the digraph of Fig. 2 and transform 3-COLORABILITY to

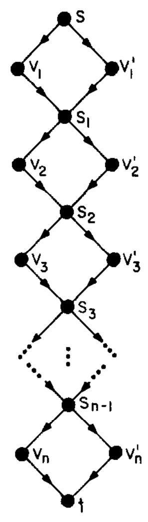

FIG. 2. The acyclic structured digraph of construction-1.

MPR. We specify the following set of must pairs in G: $M_{pr} = \{[v_i, v_j'] \mid \{v_i, v_j \in E'\}$. Then G' is 3-colorable if and only if there exists a legal path cover of size 3 for the vertices of G. The proof is similar to that of Theorem 1.

Next we consider the impossible path (IPTH) problem. In [1] it was shown that IPTH is *NP*-complete for general acyclic digraphs. We will show that IPTH is *NP*-complete even if G is an acyclic structured digraph by transforming the 3-dimensional matching (3-DM) problem to IPTH.

3-DM [4].

INSTANCE: A set $M \subseteq W \times X \times Y$, where W, X, Y are disjoint sets having the same number q, of elements.

QUESTION: Does M contain a matching, i.e., is there a subset M' of M such that $|M'| = q$ and no two elements of M' agree in any coordinate?

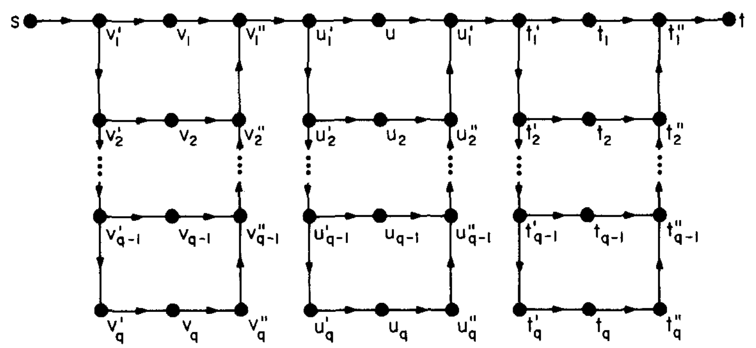

FIG. 3. The digraph G for the transformation of 3-DM to IPTH.

Given an instance of 3-DM we construct the acyclic structured digraph $G = (V, E)$ shown in Fig. 3. The digraph has $9q + 2$ vertices and thus it can be constructed in polynomial time. Then we have

THEOREM 3. *IPTH $\in$ NPC even if G is an acyclic structured digraph.*

Proof. It should be clear that IPTH $\in NP$. To show that IPTH $\in NPC$ we transform 3-DM to it. Given an instance of 3-DM we specify the following set of impossible paths in the digraph of Fig. 3:

$$\begin{aligned} I_{pth} = \{s &- v_1' - \cdots - v_i' - v_i - v_i'' - \cdots - v_1'' \\ &- u_1' - \cdots - u_j' - u_j - u_j'' - \cdots - u_1'' \\ &- t_1' - \cdots - t_k' - t_k - t_k'' - \cdots - t_1'' \mid \\ &(w_i, x_j, y_k) \in W \times X \times Y \text{ and } (w_i, x_j, y_k) \notin M\}, \end{aligned}$$

where the paths are described as vertex sequences for brevity.

We now show that M has a matching of size q if, and only if, there is a legal path cover of size q for the vertices of G. Let $M' = \{(w_{i_1}, x_{j_1}, y_{k_1}), (w_{i_2}, x_{j_2}, y_{k_2}), \ldots, (w_{i_q}, x_{j_q}, y_{k_q})\} \subseteq M$ be a matching for M. We construct a path cover $P = \{p_1, p_2, \ldots, p_q\}$ for the vertices of G by letting

$$\begin{aligned} p_m = s &- v_1' - \cdots - v_{i_m}' - v_{i_m} - v_{i_m}'' - \cdots - v_1'' \\ &- u_1' - \cdots - u_{j_m}' - u_{j_m} - u_{j_m}'' - \cdots - u_1'' \\ &- t_1' - \cdots - t_{k_m}' - t_{k_m} - t_{k_m}'' - \cdots - t_1'' \end{aligned}$$

for $m = 1, 2, \ldots, q$. Clearly P is a legal path cover for V. Conversely, given a legal path cover $P = \{p_1, p_2, \ldots, p_q\}$ for V we obtain a matching M' for M as follows:

$$M' = \{(w_{i_m}, x_{j_m}, y_{k_m}) \mid p_m \text{ visits } v_{i_m}, u_{j_m}, t_{k_m}\}.$$

Since the triples not in M correspond to impossible paths in G, it follows that $M' \subseteq M$. Since P is a path cover for V, all vertices are included in P and therefore all elements in X, Y, W are included in M'. Hence, M' is a matching of size q for M.

We have shown that IPR, IPTH, and MPR are *NP*-complete even if G is an acyclic structured digraph. If we restrict G to be a rooted tree, all three problems become trivial.

Consider next the must path problem. We will consider two alternative ways of specifying must paths. In MPTH-1 we let a must path $p(v_i, v_j)$ be a path in the digraph such that (for external reasons) if an s–t path visits v_i then it is required to contain $p(v_i, v_j)$ as a subpath [1]. Note that if an s–t path visits v_i it will always cover all the vertices in $p(v_i, v_j)$. As shown in [1], we can treat $p(v_i, v_j)$ as a single vertex v_{ij}, so that all edges directed into v_i are now directed into v_{ij}, all edges directed out of v_j are now directed out of v_{ij}, the intermediate vertices of $p(v_i, v_j)$ are removed and the edges incident on these intermediate vertices are replaced by new edges so that the reachability relations of the original digraph are preserved. Then the must path problem reduces to the problem of finding a minimum path cover for the vertices of an acyclic digraph which can be solved in polynomial time.

We may also specify must paths of the form $p((v_i, v_j), v_k)$, i.e., if an s–t path contains the edge (v_i, v_j) then it must contain $p((v_i, v_j), v_k)$ as a subpath. Let MPTH-2 be the resulting decision problem.

MPTH-2.
INSTANCE: Acyclic digraph G, set of must paths $M_{pth} \subseteq \{p((v_i, v_j), v_k) \mid (v_i, v_j) \in E\}$ and positive integer L.
QUESTION: Is there a legal path cover of size L for the vertices of G?

We show that MPTH-2 is NP-complete by transforming the 3-DM problem to it.

THEOREM 4. MPTH-2 $\in NPC$.

Proof. Clearly MPTH-2 $\in NP$. We give a polynomial transformation of 3-DM to MPTH-2. Let $M = \{(w_{i_z}, x_{j_z}, y_{k_z}) \mid 1 \leqslant z \leqslant m\}$ be an instance of 3-DM. We construct the digraph G shown in Fig. 4 and specify the following set of must paths in G:

$$\begin{aligned} M_{pth} = \{p((s, c_z), t) \mid 1 \leqslant z \leqslant m \text{ and } p(s, t) \text{ visits vertices} \\ s, c_z, s_1, v_{i_z}, s_2, u_{j_z}, s_3, w_{k_z}, t \text{ in } G\} \\ \cup \{p((s, \beta), t) \mid p(s, t) \text{ visits vertices} \\ s, \beta, c_1, c_2, \dots, c_m \text{ and } t \text{ in } G\}. \end{aligned}$$

Note that the must paths in the first subset of M_{pth} correspond to the members of M. Clearly, G and M_{pth} can be constructed in polynomial time. We next show that M has a matching of size q if, and only if, there is a legal path cover of size less than or equal to $q + 1$ for the vertices of G.

Let $M' = \{(w'_{i_1}, x'_{j_1}, y'_{k_1}), \dots, (w'_{i_q}, x'_{j_q}, y'_{k_q})\} \in M$ be a matching for M. We construct a path cover $P = \{p_1, p_2, \dots, p_{q+1}\}$ for the vertices of G as follows:

$$p_z = s - c_z - s_1 - v'_{i_z} - s_2 - v'_{j_z} - s_3 - v'_{k_z} - t \qquad \text{for} \quad 1 \leqslant z \leqslant q,$$

and

$$p_{q+1} = s - \beta - c_1 - c_2 - \cdots - c_m - t,$$

where the paths are given as vertex sequences for brevity. Clearly, P is a legal path cover of size $q + 1$ for the vertices of G.

Conversely, suppose that there exists a legal path cover of size L, $L \leqslant q + 1$ in G. Since any legal path that visits vertex β can not visit any of the vertices $v_{i1}, v_{i2}, \dots, v_{iq}$ and since no path can visit more than one v-vertex, it follows that any legal path cover in G must contain exactly $q + 1$ paths. In a legal path cover, the q paths that do not visit vertex β must each visit one v-vertex, one u-vertex, and one w-vertex in G.

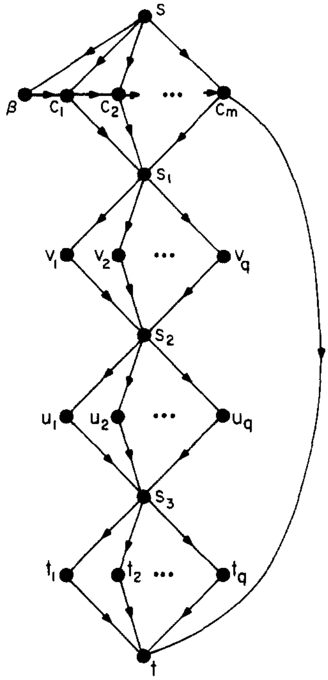

FIG. 4. The digraph G for the transformation of 3-DM to MPTH-2.

Furthermore, these three vertices must correpond to a triple in M. Then, the set of triples in M that correspond to the q paths in the legal path cover in G that do not visit β, forms a matching of size q for M.

We showed that MPTH-2 is NP-complete. If G is an acyclic structured digraph we note that in a must path of the form $p((v_i, v_j), v_k)$, either v_i dominates v_j or v_j dominates v_i. Then, the must path $p((v_i, v_j), v_k)$ is equivalent to one of the must paths $p(v_i, v_k)$ or $p(v_j, v_k)$. Then, we can replace the must paths in MPTH-2 with equivalent ones and construct an instance of MPTH-1 which can be solved in polynomial time.

3. *Subgraph Covers*

Let $S = \{V_1, V_2, \ldots, V_k\}$ be a set of vertex subsets of an acyclic digraph $G = (V, E)$ and consider the problem of covering a representative from each subset. A set of s–t paths P covers S if, for every subset V_i of S there exists at least one vertex $v_j \in V_i$ that is visited by some path in P. Then we have the following problem:

> **SUBCOVER.**
> **INSTANCE:** Acyclic digraph G, set of vertex subsets S, positive integer k.
> **QUESTION:** Is there a path cover of size m for the vertex subsets in S?

We will show that SUBCOVER $\in NPC$ even if G is restricted to be a binary rooted tree. Clearly SUBCOVER $\in NP$ as we can arbitrarily select k paths in a rooted tree and determine in polynomial time whether or not every subset is covered. To show that SUBCOVER $\in NPC$ we will transform a version of the VERTEX COVER problem to it.

VERTEX COVER [4].
INSTANCE: Graph $G=(V,E)$ with no vertex degree exceeding 4, positive integer $k \leqslant |V|$.
QUESTION: Is there a subset $V' \subseteq V$ with $|V'|=k$ such that for each edge $(v_i, v_j) \in E$ at least one of v_i, v_j belongs to V'?

Given an instance of VERTEX COVER we construct an instance of SUBCOVER by constructing a rooted tree T, where a vertex v_i with degree d_i in G corresponds to a path from the root to a leaf of T and the last d_i vertices of this path correspond to the d_i edges incident on v_i. Figure 5 illustrates the construction. Note that because no vertex in G has degree greater than four, the tree has at most $\lceil \log(|V|) \rceil + 4$ levels or at most $32|V|$ vertices and thus the construction is polynomial. The set of vertex subsets specified for T is $S = \{\{v_i, v_j\} \mid v_i, v_j$ correspond to the same edge in $G\}$. Then we have

THEOREM 5. *There is a vertex cover of size k in G if, and only if, there is a path cover of size k for the vertex subsets of S.*

Proof. Given a vertex cover of size k in G we construct a path cover for S by selecting the k paths in T that correspond to the vertices in the vertex cover. Clearly, these paths will cover all subsets in T.

Conversely, suppose that there exists a path cover of size k for the subsets in T. We can partition the paths in the path cover into two disjoint subsets P_1 and P_2,

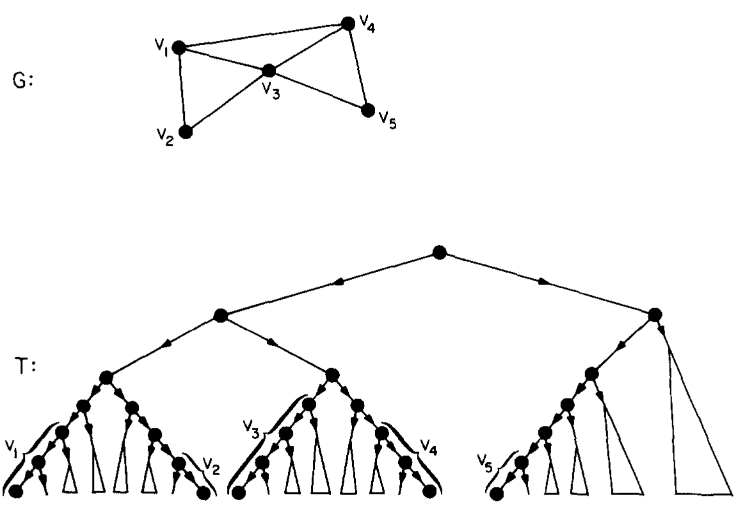

FIG. 5. The transformation of VERTEX-COVER to SUBCOVER.

those that correspond to vertices in G and those that do not respectively. Let $p \in P$. If $p \in P_1$, we include the corresponding vertex in G in the vertex cover. If $p \in P_2$, we can always find a path p' that corresponds to a vertex in G and covers the same subsets as p. Then we include the vertex corresponding to p' in the vertex cover. The set of vertices thus obtained will have cardinality k and clearly is a vertex cover since covering the subsets of S is equivalent to covering the edges of G.

Consider next the SUBCOVER problem when we require that each vertex subset V_i induces a connected digraph in G. The digraph induced by V_i, denoted $\langle V_i \rangle$, is the subgraph of G obtained by removing from G all vertices not in V_i and all edges with an endpoint outside V_i. We will refer to the problem of finding a path cover in this case as CON-SUBCOVER. To show that CON-SUBCOVER is *NP*-complete we use the *NP*-complete problem known as THREE-SATISFIABILITY (3-SAT).

> **3-SAT** [4].
> **INSTANCE**: A set U of variables, collection C of clauses over U such that each clause contains three literals, no variable appears more than once in a clause and every literal appears at least once in C.
> **QUESTION**: Is there a satisfying truth assignment for C (i.e., is there an assignment of binary values "1", "0" to the variables in U so that $C = 1$)?

THEOREM 6. CON-SUBCOVER $\in$ *NPC*.

Proof. Clearly, CON-SUBCOVER $\in$ *NP*. To show that it is *NP*-complete we will transform 3-SAT to it. Let k be the number of variables in C and n the number of clauses. Given an instance of 3-SAT we produce an instance of CON-SUBCOVER as follows. We construct the acyclic digraph $G = (V, E)$, where $V = \{v_{i1}, v_{i2} \mid 1 \leqslant i \leqslant k\} \cup \{s, t\}$ and $E = \{(s, v_{11}), (s, v_{12})\} \cup \{(v_{ij}, v_{1m}) \mid 1 \leqslant i < 1 \leqslant k, 1 \leqslant j, m \leqslant 2\} \cup \{(v_{k1}, t), (v_{k2}, t)\}$. Vertices v_{i1}, v_{i2} correspond to literals $x_i, \bar{x}_i$, respectively. Then we specify the set of vertex subsets $S = \{$vertex triples that correspond to the literals in a clause of $C\}$. By construction, these triples induce connected subgraphs. Then, C is satisfiable if, and only if, there is a path cover of size one for the vertex subsets in S. Given a satisfying assignment for C we construct a path that covers S as follows: The path visits exactly those vertices that correspond to the literals assigned to value "1," i.e., if $x_i = 1$ then the path visits v_{i1}, if $x_i = 0$ then the path visits v_{i2}. Clearly this path covers all vertex subsets since every clause in C will contain at least one literal assigned the value "1." Conversely, given a path that covers all the vertex subsets in S we obtain a satisfying assignment by letting $x_i = 1$ if the path visits v_{i1} or letting $x_i = 0$ if the path visits v_{i2}. If the path visits neither v_{i1} nor v_{i2} we make an arbitrary assignment to x_i. The resulting assignment satisfies C since the path covers all triples corresponding to the clauses of C, i.e., at least one literal in each clause is assigned the value "1."

If we restrict G to be an acyclic structured digraph or a rooted tree, CON-SUBCOVER can be solved in polynomial time. In a rooted tree, a vertex subset that induces a connected subgraph must induce a rooted tree and therefore there exists a vertex in the subset that dominates all other vertices in it. Then covering the vertex subset is equivalent to covering the root of the subgraph induced by the vertex subset. Similarly, in an acyclic structured digraph, each subset V_i will contain at least one vertex that dominates all vertices in V_i. This can be easily seen if we define levels of dominators in G [5]. Let the 0-dominators at G be the vertices that are visited by every s–t path in G. Then, we recursively define the i-dominators of G to be those vertices that are 0-dominators of the digraphs obtained by removing from G all vertices that are j-dominators for $j < i$. Clearly, if vertex subset V_i induces a connected subgraph, and v_j is a k-dominator with least k in V_i, every path that covers V_i must visit v_j. Thus, we can replace the vertex subsets with appropriate representative vertices and CON-SUBCOVER reduces to the problem of finding a minimum path cover for a set of vertices of G which can be solved in polynomial time.

4. *Length Constrained Path Covers*

Another class of constrained path cover problems is obtained when an upper limit is placed on the length of the paths that can be included in the path cover. Let $1(e)$ be the length of edge e and let the length of an s–t path be the sum of the lengths of the edges traversed by the path. Then we have

MAXLENGTH-COVER.
INSTANCE: Acyclic digraph $G = (V, E)$, integer lengths $1(e)$ for all $e \in E$ and integers L, M.
QUESTION: Is there a path cover P for the vertices of G such that $|P| \leqslant |M|$ and for all $p \in P$, the length of p does not exceed L?

We will show that MAXLENGTH-COVER $\in NPC$ by transforming the 3-PARTITION problem to it.

3-PARTITION.
INSTANCE: A finite set A of $3m$ elements, integer bound B and integer size $s(a)$ for each $a \in A$ with $B/4 < s(a) < B/2$ and $\sum_{a \in A} s(a) = mB$.
QUESTION: Can A be partitioned into m disjoint sets $S_1, S_2, \ldots, S_m$ such that $\sum_{a \in S_1} s(a) = B$ and $|S_i| = 3$ for $1 \leqslant i \leqslant m$?

Given an instance of 3-PARTITION we produce an instance of MAXLENGTH-COVER by constructing the acyclic structured digraph shown in Fig. 6. The digraph G consists of $3m$ similar subgraphs connected in series. The ith subgraph corresponds to the ith element of the set A. Then we have

THEOREM 7. *There is a path cover P for the vertices of G with $|P| \leqslant m$ and $1(p_i) \leqslant B + 9m - 1$ for all $p_i \in P$ if, and only if,* 3-PARTITION *has a solution.*

Proof. Given a partition $A = S_1 \cup S_2 \cup \cdots \cup S_m$ such that $\sum_{a \in S_i} s(a) = B$ we construct a legal path cover for the vertices of G as follows. We use m paths each corresponding to a subset S_i, $1 \leqslant i \leqslant m$. Path p_i visits the three edges corresponding to the members of S_i in the appropriate subgraphs of G. In the remaining subgraphs,

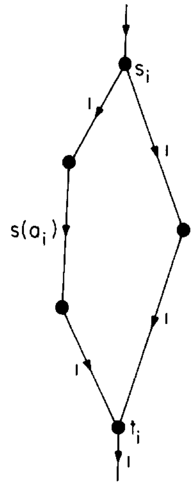

FIG. 6. The subgraph G corresponding to the ith element of S in the transformation from 3-PARTITION to MAXLENGTH-COVER.

p_i will visit the vertex of degree two that is not associated with a member of A. The length of p_i is $(s(a_{i1}) + s(a_{i2}) + s(a_{i3})) + 2(3m) + (3m - 1) = B + 9m - 1 \leqslant L$.

Conversely, suppose that we are given a legal path cover P for the vertices of G with $|P| = m$. Note that no path $p_i \in P$ can visit more than three edges associated with a member of A (if it does, its length is at least $B + 1 + 9m - 1 = L + 1 > L$). Then, since there are $3m$ edges associated with members of A and each path cannot visit more than three such edges, it follows that each path $p_i \in P$, visits exactly three edges associated with members of A. Also, we have that the sum of the lengths of these three edges must be $L - 2(3m - 3) - 6 - (3m - 1) = B$, which means that the weights of the corresponding members of A sum up to B. Thus, the m paths in the path cover produce a 3-partition for A.

COROLLARY. MAXLENGTH-COVER $\in$ *NPC even if G is an acyclic structured digraph.*

Proof. The digraph described in Fig. 6 is an acyclic structured digraph. Then the result follows from Theorem 6.

Consider next the MAXLENGTH COVER problem when $1(e) = 1$ for all edges in the digraph. Then we have

COROLLARY. MAXLENGTH COVER $\in$ *NPC even of G is an acyclic structured digraph and* $1(e) = 1$ *for all* $e \in E$.

Proof. The 3-PARTITION problem is *NP*-complete in the strong sense, that is it remains *NP*-complete even if the sizes $s(a)$ are polynomially related to the number of elements in A [4]. Then we can modify the digraph of Fig. 6 so that an edge of length

x is replaced by x edges of unit length by adding the appropriate number of vertices of degree two. Since, the maximum length in G is polynomially related to $|A|$, the new digraph G can be constructed in polynomial time. Then, Theorem 6 holds for G and MAXLENGTH-COVER $\in NPC$ even if all edges have unit length.

If G is a rooted tree, the MAXLENGTH COVER problem reduces to finding the longest path in the tree which can be done in polynomial time.

III. The General Problem

In the previous section we have considered a number of path cover problems and studied their complexity. In this section we look at path cover problems in general and point out a relation between the complexity of such problems and the existence of a reachability relation on the elements that are to be covered.

Let $X = \{x_1, x_2, ..., x_m\}$ be a set of elements in an acyclic digraph. The members of X may be vertices, edges, required pairs, etc. Element x_i is said to be path-related to element x_j ($x_i R_p x_j$) if there exists an s–t path in G that covers both x_i and x_j. Then, the problem of finding a path cover for X can be stated as follows:

PATH-COVER.

INSTANCE: Acyclic digraph G, set X of elements of G, a path relation R_p on elements of G and positive integer k.

QUESTION: Is there a path cover of size k for the elements in X, i.e., is there a set of paths $P = \{p_1, ..., p_k\}$ such that for all $x_i \in X$, there is a path $p_j \in P$ that covers x_i and for any two elements x_i, $x_j \in X$ such that x_i, $x_j \in p_i$ for some $p_i \in P$, we have that $x_i R_p x_j$?

Clearly, the problems discussed in the previous section are special cases of PATH-COVER. Then we have

Corollary. PATH-COVER $\in NPC$.

Proof. PATH-COVER $\in NP$ since we can arbitrarily select k paths in G and determine in polynomial time whether or not they constitute a path cover for X. Since REQPR is a special case of PATH-COVER, if PATH-COVER $\in P$ then REQPR $\in P$. But REQPR $\in NPC$ and therefore PATH-COVER $\in NPC$.

In examining the path cover problems for which it is known that polynomial time algorithms exist (e.g., MPTH-1, REQPTH) we note that they reduce to the problem of finding a path cover for the vertices of an acyclic digraph. This reduction is made possible because the path relation R_p can be replaced by a reachability relation (partial order) on the elements that are to be covered. In general, suppose that a reachability relation on the elements of X can be defined in polynomial time. Then, we have

Theorem 8. *If a reachability relation on the elements of X can be defined in polynomial time, then* PATH-COVER *can be solved in polynomial time.*

Proof. We find a minimum path cover for X by constructing the acyclic digraph $G' = (V', E')$, where $|V'| = |X|$ and $(v_i, v_j) \in E'$ if x_i reaches x_j in G. Then, finding a minimum path cover for X in G is equivalent to finding a minimum path cover for the vertices of G' which can be done in polynomial time.

Consider next the path cover problems that have been shown to be *NP*-complete. For each of them there is a natural definition for reachability on the elements that are to be covered. For example, in REQPR we can say that required pair $r_1 = [v_i, v_j]$ "reaches" required pair $r_2 = [v_m, v_n]$ if v_i reaches v_m (v_j reaches v_n if $v_i = v_m$) and there is an s–t path that covers both r_1 and r_2. This however, is not a valid reachability relation (i.e., a partial order) as it is not transitive. For example, in Fig. 7, required pair $[v_1, v_8]$ reaches $[v_3, v_5]$, $[v_3, v_5]$ reaches $[v_3, v_7]$, but $[v_1, v_8]$ does not reach $[v_3, v_7]$. Also note that the cardinalities of a minimum path cover and a maximum set of mutually incomparable required pairs are not necessarily equal. In Fig. 7, the cardinality of a minimum path cover for the required pairs is 3 but there is no set of three mutually incomparable required pairs. As a second example consider the SUBCOVER problem. A natural definition for reachability between vertex subsets is derived as follows. First, we assign an index to each vertex so that $i < j$ if v_i reaches v_j. Then we order the vertices in each vertex subset in order of increasing index and order the vertex subsets themselves in lexicographical order of their index sequences. Then a subset V_i reaches a subset V_j if $i < j$ and there is an s–t path that covers both V_i and V_j. Again, this definition does not lead to a valid reachability relation. In the digraph of Fig. 8, vertex subset $\{v_1, v_3\}$ reaches $\{v_1, v_6\}$, and $\{v_1, v_6\}$ reaches $\{v_2, v_4\}$, but $\{v_1, v_3\}$ does not reach $\{v_2, v_4\}$. Finally, consider the

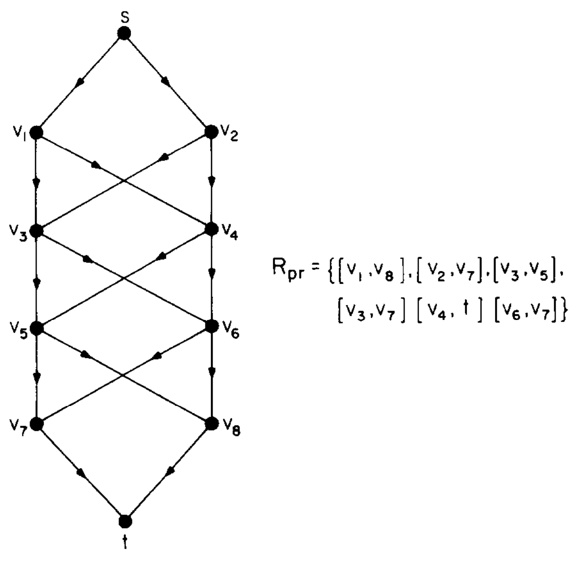

FIG. 7. Example for REQPR. A minimum path cover is $\{s - v_1 - v_3 - v_5 - v_8 - t, s - v_2 - v_4 - v_6 - v_7 - t, s - v_1 - v_3 - v_5 - v_7 - t\}$.

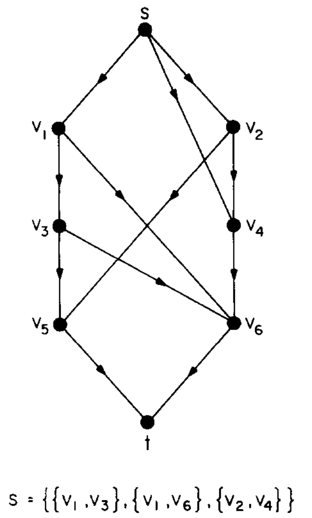

FIG. 8. Digraph for SUBCOVER.

MAXLENGTH COVER problem. We can define that a vertex v_i L-reaches vertex v_j if v_i reaches v_j and there is an s–t path of length no greater than L that visits both v_i and v_j. Again, L-reachability is not a partial order. In the digraph of Fig. 9, let $L = 3$. Then v_1 L-reaches v_2, v_2 L-reaches v_3, but v_1 does not L-reach v_3. Similarly, in all other path cover problems that were shown to be NP-complete, what seems to be the natural way to define reachability on the elements that are to be covered does not produce a partial order. Thus, there appears to be a strong connection between the NP-completeness of a path cover problem and the existence of a valid reachability relation on the elements that are to be covered. Then we propose the following conjecture:

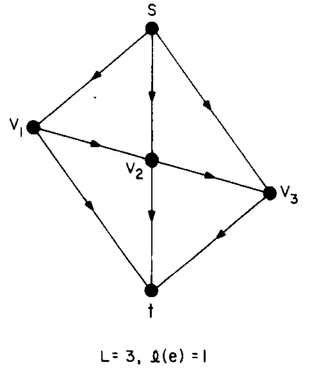

FIG. 9. Example for MAXLENGTH-COVER.

Conjecture. A path cover problem is *NP*-complete if, and only if, no valid reachability relation (partial order) on the elements that are to be covered can be defined in polynomial time (assuming $P \neq NP$).

To disprove the conjecture we need to produce a path cover problem which can be solved in polynomial time and in which no reachability relation on the elements that are to be covered can be defined in polynomial time. The main difficulty in producing a counter example to the conjecture is pointed out by the MPTH-1 problem. To reiterate, in MPTH-1 we are looking for a legal path cover for the vertices of an acyclic digraph where a path p is legal if for any must path $p(v_i, v_j)$ if $v_i \in p$ *then* $p(v_i, v_j)$ is a subpath of p. We can define reachability as follows: Vertex v_i *m*-reaches v_j if v_i reaches v_j and there is a legal *s–t* path that visits both v_i and v_j. This relation is not a partial order (see Fig. 10). However, note that for each must path $p(v_i, v_j)$, the initial vertex v_i in a sense dominates all other vertices in $p(v_i, v_j)$ since if a path covers v_i it necessarily covers all other vertices in $p(v_i, v_j)$. Since v_i is one of the elements to be covered we can disregard the remaining vertices in $p(v_i, v_j)$. That is, we can replace a must path $p(v_i, v_j)$ with its initial vertex v_i by an appropriate modification of the edge set. Then, a reachability relation can be easily defined and as was shown in [1], MPTH-1 can be solved in polynomial time. In the case of MPTH-1 it is fairly easy to modify the problem so that a valid reachability relation can be defined. However, this may not always be the case and that complicates the search for a counterexample.

Let P be a minimum path cover for the set X of elements of an acyclic digraph in a path cover problem A and let I be a maximum set of mutually incomparable vertices with respect to an appropriate definition of reachability on the elements of X in problem A. Then, if the reachability relation is a partial order we have that $|P| = |I|$, i.e., a theorem similar to Dilworth's holds for problem A. If the reachability relation is not a partial order we can only state that $|P| \geqslant |I|$.

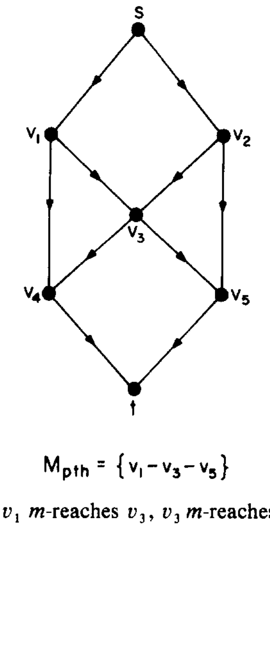

FIG. 10. In the digraph above, v_1 *m*-reaches v_3, v_3 *m*-reaches v_4, but v_1 does not *m*-reach v_4.

IV. CONCLUSIONS

We have considered a number of path cover problems in acyclic digraphs and pointed out a connection between the complexity of such a problem and the existence of a valid reachability relation on the elements that are to be covered in the digraph. Table I summarizes the complexity results for path cover problems. The optimization problems (i.e., finding minimum path covers) corresponding to the *NP*-complete problems of Table I, are *NP*-hard. In practice, one may be interested in near optimal path covers. Reductions similar to the ones used in the proof of the *NP*-completeness of IPR and MPR can be used to show that the approximation problems for IPR, MPR (i.e., find a path cover P such that $|P| \leqslant c\,|P_{\text{optimum}}|$, for any constant c, $1 \leqslant c < 2$) are also *NP*-hard. Also, the problems IPR, IPTH, MPR, and MAXLENGTH COVER remain *NP*-complete if we require a path cover for the edges (instead of the vertices) of the digraph. This follows from the fact that in an acyclic structured digraph a path cover for the vertices is also a path cover for the edges of the digraph.

We conjecture that a path cover problem is *NP*-complete if, and only if, a valid reachability relation (i.e., partial order) on the elements to be covered can be defined in polynomial time. If this conjecture is shown to be true, we would have a characterization of the complexity of path cover problems in acyclic digraphs. A shortcoming of this characterization is that the reachability relation in question is not specified precisely. As pointed out in the case of the MPTH-1 problem, although the natural definition for reachability on elements that are to be covered may not be valid, it may be possible to perform a polynomial time transformation that reduces a path cover problem into an equivalent problem for which a valid reachability relation can be defined. Further research is needed to clarify the connection between the existence of a valid reachability relation and the complexity of path cover problems.

TABLE I

Digraph G is:	Acyclic	Structured Acyclic	Rooted Tree
REQPR	*NPC*	*NPC*	*P*
REQPTH	*P*	*P*	*P*
IPR	*NPC*	*NPC*	*P*
IPTH	*NPC*	*NPC*	*P*
MPR	*NPC*	*NPC*	*P*
MPTH-1	*P*	*P*	*P*
MPTH-2	*NPC*	*P*	*P*
SUBCOVER	*NPC*	*NPC*	*NPC*
CON-SUBCOVER	*NPC*	*P*	*P*
MAXLENGTH COVER	*NPC*	*NPC*	*P*

REFERENCES

1. S. C. NTAFOS AND S. L. HAKIMI, On path cover problems in digraphs and applications to program testing, *IEEE Trans. Software Engrg.* **SE-5**, (1979), 520–529.
2. R. P. DILWORTH, A decomposition theorem for partially ordered sets, *Ann. of Math. (2)* **51**, No. 1 (1950), 161–166.
3. A. NIJENHUIS AND H. S. WILF, "Combinatorial Algorithms," Academic Press, New York, 1975.
4. M. R. GAREY AND D. S. JOHNSON, "Computers and Intractability: A Guide to the Theory of *NP*-Completeness," Freeman, San Francisco, 1978.
5. S. C. NTAFOS AND S. L. HAKIMI, On structured digraphs and program testing, *IEEE Trans. Comput.* **C-30**, No. 1 (1981), 67–77.
6. K. A. KRAUSE, R. W. SMITH, AND M. A. GOODWIN, Optimal software test planing through automated network analysis, *in* "Proceedings, 1973 IEEE Sympos. on Computer Software Reliability, New York," pp. 18–22, 1973.
7. H. N. GABOW, S. N. MAHESHWARI, AND L. J. OSTERWEIL, On two problems in the generation of program test paths, *IEEE Trans. Software Engrg.* **SE-2** (1976), 227–231.