RAIRO. RECHERCHE OPÉRATIONNELLE

MACIEJ M. SYSLO

Optimal constructions of event-node networks

RAIRO. Recherche opérationnelle, tome 15, nº 3 (1981), p. 241-260
<http://www.numdam.org/item?id=RO_1981__15_3_241_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 15, n° 3, août 1981, p. 241 à 260)

OPTIMAL CONSTRUCTIONS OF EVENT-NODE NETWORKS (*) (1)

by Maciej M. SYSLO (2)

Abstract. — *There are two types of networks in the scheduling and planning which represent a project i. e., the activities together with their precedence relations, namely, the activity networks and the event networks. For each project, there exists a unique activity network without redundant arcs but since there is an infinite number of different sized event networks, the problem is to find an event network with the minimum number of dummy activities. The motivation behind this problem is to minimize the time of the analysis of a network which is proportional to the number of activities, including the dummy ones. Krishnamoorthy and Deo proved that this problem is NP-complete. In sections 2 and 3 we characterize activity networks for which there exist event networks without dummy activities and we show that the question whether a given activity network requires dummy activities in the event network can be answered in polynomial time. We review some algorithms for finding an optimal event network and a new approach is presented which gives rise to an approximate algorithm and can lead to an optimal branch-and-bound method. Some generalizations of this real-world problem are also considered.*

Key words à phrases: Complexity, line digraph, network construction, network analysis (*A.M.S. classification:* 05C20, 05C35, 68C25, 68E10, 90C35; *C.R. categories:* 5.32).

Résumé. — *Il existe dans l'ordonnancement et la planification deux types de réseaux qui représentent un projet, c'est-à-dire les activités avec leurs relations de précédence : les réseaux d'activités et les réseaux d'événements. Pour chaque projet, il existe un seul réseau d'activité sans arc redondant, mais, puisqu'il y a un nombre infini de réseaux d'événements de tailles différentes, le problème est de trouver le réseau d'événements avec le plus petit nombre d'activités artificielles. La motivation de ce problème est de minimiser le temps d'analyse d'un réseau, temps qui est proportionnel au nombre des activités, artificielles incluses. Krishnamoorthy et Deo ont montré que ce problème est NP-complet. Dans les paragraphes 2 et 3 nous caractérisons les réseaux d'activités pour lesquels il existe des réseaux d'événements sans activités artificielles et nous montrons que l'on peut répondre en un temps polynomial à la question de savoir si un réseau donné d'activités exige des activités artificielles. Nous passons en revue quelques algorithmes pour trouver un réseau optimal d'événements, et nous présentons une nouvelle approche qui amène à un algorithme approché et peut conduire à une méthode arborescente optimale. Nous considérons également quelques généralisations de ce problème concret.*

(*) Received May 1980.

(1) The first draft of this paper has been written when the author was with University of Tokyo as a Mombusho Scholarship Student in 1975-1976 and it has been completed when he visited Washington State University as a Visiting Computer Scientist in July 1979.

(2) Institute of Computer Science, University of Wroclaw, Pl. Grunwaldzki 2/4, 50-384 Wroclaw, Poland.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/1981/241/$ 5.00

1. INTRODUCTION

In the scheduling and planning, there are two types of networks which represent a *project*, i. e., the activities together with their precedence relations, namely the *activity-node networks* and the *event-node networks*. The former are sometimes called simply the *activity* networks and the latter the PERT, *project*, or *event* networks. In this paper, we shall use the names activity network and event network, resp. An activity network is a digraph D in which the nodes correspond one-to-one with the given activities and there is an arc (u, v) in D if activity u precedes activity v. There exists a unique activity network without redundant arcs for each project. In an event network E which corresponds to an activity network D, the given activities are represented by a subset of arcs of E and the precedence relations are preserved. In general, *dummy* activities (arcs of E) are introduced to satisfy the last requirement and, since there is an infinite number of different sized event networks for each project, the problem is to find for a set of activities and their precedence relations, an event network with the minimum number of dummy activities. The motivation behind this problem is to minimize the time of the analysis of a network which is proportional to the number of arcs, including those which correspond to dummy activities.

Krishnamoorthy and Deo proved in [9] that the problem of finding the minimum number of dummy activities in the event network which correspond to a given set of activities and their precedence relations is NP-complete. In section 2 and 3, we characterize the precedens relations for which there exists an event network without dummy activities and show that the question whether a given precedence relations require dummy activities in the event network can be answered in polynomial time.

In section 4-6, we review some algorithms for finding the event network with the minimum number of dummy activities and in section 7, a new approach is presented which gives rise to an approximate algorithm and can lead to an optimal branch-and-bound method.

The precedence relations of a real-world set of activities are consistant, that is the corresponding activity network and the event network contain no circuit. Cantor and Dimsdal [2] generalized the problem for not necessarily acircuit digraphs and we investigate and explore here some graph-theoretic relations between two pairs of digraphs, namely between an activity network and its event network and a digraph and its line digraph.

For graphical terms not defined in this paper we referred to [7].

2. DEFINITIONS. THE PROBLEM

Let $D=(V, A)$ denote a *directed graph* (simply, *digraph*), where V is the set of *nodes* and A is the set of *arcs*, that is, ordered pairs of nodes. An arc is denoted by (u, v). To avoid misunderstandings, the set of nodes and the set of arcs of D are sometimes denoted by $V(D)$ and $A(D)$. Let us define

$$\Gamma_D\, v=\{u\in V(D) : (v, u)\in A(D)\}$$

and

$$\Gamma_D^{-1}\, v=\{u\in V(D) : (u, v)\in A(D)\},$$

where $v\in V(D)$. Notice that we allow D to have *loops*, that is, arcs of the form (u, u). If in addition we allow D to have *parallel* arcs that is, arcs which connect the same nodes, then D is called a *multidigraph* and A should be considered as a family of pairs.

If $u_0, u_1, \ldots, u_k$ $(k\geqq 1)$ are nodes of D and $(u_{i-1}, u_i)\in A(D)$ for $i=1, 2, \ldots, k$ then we denote $u_0\to u_k$ and say that there exists a path from u_0 to u_k. A digraph D is *acircuit* if it contains no $u_0\to u_0$. Notice that an acircuit digraph has no loops. Let $a_1=(u_0, v)$ and $a_l=(w, u_l)$. If $v=w$ or $v\to w$ then we denote $a_1\to a_l$.

Let D be an activity network of a given project which consists of a set of activities and precedence relations among them, that is there exists a one-to-one correspondence between the nodes of D and the activities, and $(u, v)\in A(D)$ if activity u precedes activity v. D is an acircuit digraph. The problem of constructing an event network for D with the minimum number of dummy activities is to find a digraph E such that (1) there exists a one-to-one correspondence $\alpha : V(D)\to B$, where $B\subseteq A(E)$ such that $u\to v$ in D if and only if $\alpha(u)\to\alpha(v)$ in E for any $u, v\in V(D)$, and (2) the set of *dummy* activities (arcs) $A(E)-B$ has the minimum number of elements among all digraphs which satisfy (1).

Cantor and Dimsdal [2] dealt with the problem for digraphs which are not necessarily acircuit. Let D be a digraph. The pair (E, f), where E is a digraph and $f : V(D)\to A(E)$ is an *arc-dual* digraph of D if for any pair $u_1, u_2\in V(D)$, we have $u_1\to u_2$ in D if and only if $f(u_1)\to f(u_2)$ in E. The arcs in $A(E)-f(V(D))$ are called *dummy arcs* of E. If E is a digraph, then the pair (D, g), where D is a digraph and $g : A(E)\to V(D)$ is a *node-dual* digraph of E if for any pair $a_1, a_2\in A(E)$ we have $a_1\to a_2$ in E if and only if $g(a_1)\to g(a_2)$ in D. It is easy to see that in the class of acircuit digraphs, D may be considered an activity network and E a corresponding event network.

Let S be a finite set. The family of subsets $\{S_i\}_I$ of S some of which may be empty is called an *improper cover* of S if $\bigcup_{i\in I} S_i=S$. If additionally for every $k, l\in I$, if $k\neq l$ then $S_k\cap S_l=\emptyset$, then $\{S_i\}_I$ is called an *improper partition* of S. Let F be a digraph. Then $\langle\{U_i\}, \{W_i\}\rangle_I$, where $\{U_i\}_I$ and $\{W_i\}_I$ are improper covers (partitions) of $V(F)$ is called an *improper cover* (*partition*) of a digraph F if $A(F) = \bigcup_{i \in I} U_i \times W_i$, where $\times$ denotes the cartesian product of sets. Every digraph D has an improper cover, for instance if $A(D) = \{e_1, e_2, \ldots, e_m\}$ and $e_i = (u_i, v_i)$ then $U_i = \{u_i\}_I$ and $W_i = \{v_i\}_I$, where $I = \{1, 2, \ldots, m\}$, *see* also theorem 2.1.

We shall also consider line digraphs. If E is a multidigraph then the *line digraph* $\mathscr{L}(E)$ of E is defined as follows $V(\mathscr{L}(E)) = A(E)$ and if $a_1 = (u_1, v_1)$, $a_2 = (u_2, v_2)$, $a_1, a_2 \in A(E)$ then $(a_1, a_2) \in A(\mathscr{L}(E))$ if and only if $v_1 = u_2$. A digraph D is said to be a *line digraph* or *reversible* if there exists a multidigraph E such that $D = \mathscr{L}(E)$. There exist several characterizations of line digraphs and here we shall make use of the following.

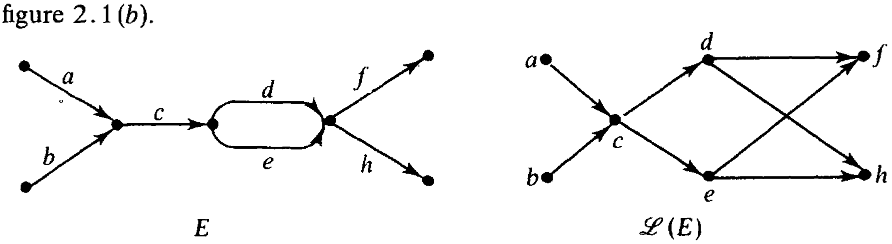
THEOREM 2.1 [6]: *A digraph D is a line digraph if and only if there exists an improper partition of D.* □

It is easy to show that the last theorem is equivalent to the following.

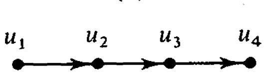
THEOREM 2.2 [12]: *A digraph D is a line digraph if and only if $\Gamma_D v_1 \cap \Gamma_D v_2 \neq \emptyset$ then $\Gamma_D v_1 = \Gamma_D v_2$, where $v_1, v_2 \in V(D)$.* □

COROLLARY TO THEOREM 2.2: *A digraph D is a line digraph if and only if there exists an improper partition $\{V_j\}_J$ of $V(D)$ such that for each $v \in V(D)$ there exists $k \in J$ such that $\Gamma_D v = V_k$.* □

If E is a multidigraph then $\mathscr{L}(E)$ is a node-dual digraph of E. In this case, g is a bijection and $\mathscr{L}(E)$ need not be a node-dual digraph of E with the minimum number of vertices. Figure 2.1 (a) shows a digraph E and its line digraph, and the node-dual digraph of E with the minimum number of nodes is shown in figure 2.1 (b).



(a)



$g(a) = g(b) = u_1$, $g(c) = u_2$, $g(d) = g(e) = u_3$, $g(f) = g(h) = u_4$

(b)

Figure 2.1.

If D is a line digraph, i. e., when there exists a digraph E such that $\mathscr{L}(E)=D$ then (E, h^{-1}) is an arc-dual digraph of D and in this case h^{-1} is a bijection between $V(D)$ and $A(E)$, that is, E contains no dummy arcs. Figure 2.2 (*a*) shows a digraph D which is not a line digraph and for which there exists the arc-dual digraph (E, f) without dummy arcs *see* figure 2.2 (*b*) [f_i denotes $f(u_i)$]. Notice, that $\mathscr{L}(E)$ is isomorphic to D', where $V(D')=V(D)$ and

$$A(D')=A(D)\cup\{(u_2, u_3)\}-\{(u_1, u_4)\}.$$

Since our goal is to minimize the number of dummy activities in an arc-dual digraph (E, f) of D, first we should characterize a digraph D such that there exists a map f which maps $V(D)$ onto $A(E)$ (f is not necessarily to be bijective).

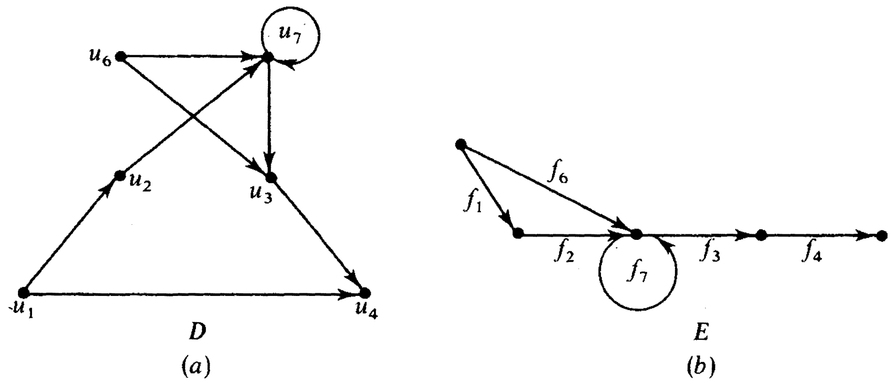


Figure 2.2.

If D is a digraph then let $\check{D}$ denote the *transitive closure* of D, i. e., $V(\check{D})=V(D)$ and $(u, v)\in A(\check{D})$ if and only if $u\to v$ in D. A digraph D' is the *transitive reduction* of D if (1) D' is a spanning subdigraph of D, i. e.:

$$V(D')=V(D) \text{ and } E(D')\subseteq E(D),$$

(2) $\check{D}'=\check{D}$, and (3) D' has the minimum number of arcs among all digraphs which satisfy (1) and (2).

THEOREM 2.3: *A digraph D has an arc-dual digraph (E, f) without dummy arcs if and only if there exists a digraph D' such that D' is a line digraph and $\check{D}=\check{D}'$.*

Proof: If for a digraph D there exists an arc-dual digraph (E, f) such that f maps $V(D)$ onto $A(E)$ then we can construct the digraph D' such that D' is a line digraph, (E, f) is also an arc-dual digraph of D', and $\check{D}'=\check{D}$. Let $V(D')=V(D)$. Then $(u_1, u_2)\in A(D')$ if and only if, if $f(u_1)=(v_1, w_1)$ and $f(u_2)=(v_2, w_2)$ then

$w_1 = v_2$. It is easy to verify that D' is a line digraph. By the construction, we have $u \to v$ in D' if and only if $f(u) \to f(v)$ in E, therefore (E, f) is also an arc-dual digraph of D' and therefore $\breve{D}' = \breve{D}$. The proof of the converse we leave to the reader as an exercise. □

In a digraph D, an arc (u, v) is *redundant* if there exists a path $u \to v$ consisting of at least three vertices. If D is an acircuit digraph then we have the following.

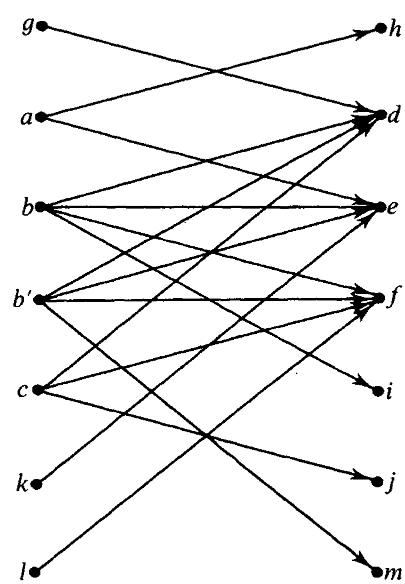
LEMMA 2.1: *If D is an acircuit line digraph then D has no redundant arcs.*

Proof: Let $u_0, u_1, \ldots, u_l \in V(D)$, $(u_{i-1}, u_i) \in A(D)$ $(i=1, 2, \ldots, l, l \geqq 2)$ and suppose that $(u_0, u_l) \in A(D)$. Hence $u_l \in \Gamma_D u_0 \cap \Gamma_D u_{l-1}$ and $u_1 \in \Gamma_D u_0 - \Gamma_D u_{l-1}$, therefore, by theorem 2.2, D is not a line digraph. □

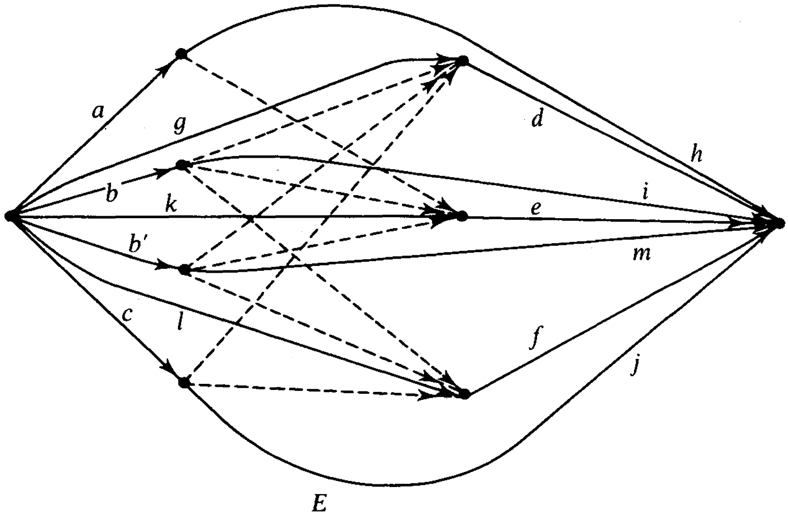
THEOREM 2.4: *If D is an acircuit digraph then D has an event network without dummy activities if and only if the transitive reduction D' of D is a line digraph.*

Proof: If D is acircuit then every digraph D' such that $\breve{D}' = \breve{D}$ is a spanning subdigraph of $\breve{D}$ and by lemma 2.1, if D' is a line digraph then D' is the transitive reduction of D. □

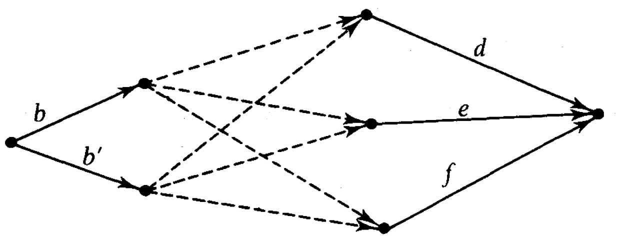
In the next section we shall use theorems 2.3 and 2.4 to show that the problem of finding whether there exists an event network without dummy activities can be solved in polynomial time.



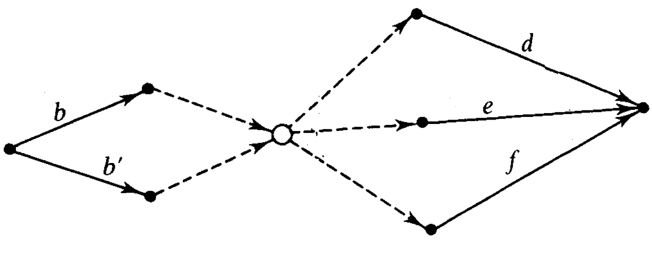
(a)



(b)



(c)



(d)

Figure 2.3.

We conclude this section with an example which shows that in spite of the results in [3] and [10], the number of nodes and the number of arcs in an event network cannot be minimized simultaneously even if there are no parallel activities (i. e., activities with the same sets of predecessors and successors).

The set of activities and their precedence relations are shown in figure 2.3 (*a*). Figure 2.3 (*b*) shows the event network *E* of *D* with the minimum number of nodes, and figures 2.3 (*c*) and 2.3 (*d*) show how to decrease the number of dummy activities in *E* by increasing the number of nodes.

3. THE COMPUTATIONAL COMPLEXITY OF THE PROBLEM

Krishnamoorthy and Deo proved in [9] that the problem of finding whether there exists the event network with the number of dummy activities less than k for a given set of activities and their precedence relations is NP-complete. In fact, they proved a stronger result, that this problem is NP-complete even if we restrict our attention only to the event networks with the minimum number of nodes. The result of Krishnamoorthy and Deo follows from the fact that the node-cover problem in simple graphs with vertices of degree two or three is polynomially transformable to the problem considered here.

Applying the results of the previous section we show now that the problem of testing whether for a given digraph not necessarily acircuit there exists an arc-dual digraph without dummy arcs, i. e., when $k=0$, can be solved in polynomial time.

Let consider first the real-world problem. The following algorithm checks whether for a given acircuit activity network D there exists an event network with no dummy activities.

ALGORITHM 3.1:

1. Find the transitive reduction D' of D.

2. If D' is a line digraph then there exists an event network of D which has no dummy activities. □

THEOREM 3.1: *Algorithm 3.1 tests in polynomial time whether an acircuit activity network D has an event network with no dummy activities.*

Proof: The correctness of the algorithm follows from theorem 2.4. Regarding the complexity, step 1 needs $O(n^3)$ time or less, where $n = |V(D)|$ (*see* [1]) and we encourage the reader to show that applying corollary to theorem 2.2, step 2 can be implemented in $O(m)$ time, where $m = |A(D)|$. □

In general, when D is an arbitrary digraph, the problem of testing whether D has an arc-dual digraph with no dummy arcs can be reduced to that for the *condensation* D^* of D. The vertices of D^* are in a one-to-one correspondence with the strongly connected components of D and $(u^*, v^*) \in A(D^*)$, where u^*, $v^* \in V(D^*)$ if and only if there exist $u, v \in V(D)$ such that u and v belong to the strongly connected components u^* and v^* of D, resp., and $(u, v) \in A(D)$.

ALGORITHM 3.2:

1. Find the condensation D^* of D.

2. If $|V(D^*)| > 1$ then remove all arcs (u_0^*, u_l^*) of D^* such that there exist u_0^*, $u_1^*, \ldots, u_l^* \in V(D^*)$, $(u_{i-1}^*, u_i^*) \in A(D^*)$ $(i = 1, 2, \ldots, l)$ and:

(a) $l > 2$, or

(b) $l = 2$, and

the component u_1^* consists of one vertex $u_1 \in V(D)$, and $(u_1, u_1) \notin A(D)$.

3. If $|V(D^*)| = 1$ or D^* is a line digraph then there exists an arc-dual digraph of D which has no dummy arcs. □

THEOREM 3.2: *Algorithm 3.2 tests in polynomial time if a given digraph has an arc-dual digraph with no dummy arcs.*

We shall use the following lemmas to prove theorem 3.2.

LEMMA 3.1: *If D is a strongly connected digraph then $\check{D}$ is a line digraph.* □

LEMMA 3.2: *Let D be a line digraph, $u_0, u_1, \ldots, u_l \in V(D)$, $(u_{i-1}, u_i) \in A(D)$ $(i = 1, 2, \ldots, l)$ $l \geqq 3$, and u_0, u_l and at least two vertices in $\{u_1, u_2, \ldots, u_{l-1}\}$ belong to different strongly connected components of D then $(u_0, u_l) \notin A(D)$.*

Proof: If $(u_0, u_l) \in A(D)$ then $u_l \in \Gamma_D u_0 \cap \Gamma_D u_{l-1}$ and $u_1 \in \Gamma_D u_0 - \Gamma_D u_{l-1}$, since u_0 and u_l and least two vertices in $\{u_1, \ldots, u_{l-1}\}$ belong to different strongly connected components of D. Therefore D is not a line digraph. □

Proof of theorem 3.2: First, notice that if a strongly connected component of D consists of at least two vertices u and v then they have the same sets of predecessors and successors and $(u, u), (v, v) \in A(\check{D})$. Therefore the problem of finding a digraph D' such that $\check{D}' = \check{D}$ can be reduced to that for the condansation D^* of D. The correctness of step 2 follows from lemmas 3.1 and 3.2, and step 3 is based on theorem 2.3 and lemma 3.1. Regarding the complexity of algorithm 3.2, the condensation D^* of D can be found in $O(m)$ time, where $m = |A(D)|$, step 2 is a partial transitive reduction of D^* therefore it can be done in time bounded by $O(n^3)$, where $n = |V(D)|$ and step 3 needs $O(m^*)$ time, where $m^* = |A(D^*)|$. □

Notice that the problem of testing whether for a given activity network there exists an event network without dummy activities is also significant from a practical point of view. The result of Krishnamoorthy and Deo suggests that a polynomial approximate algorithm rather than an exact one should be used in practice. However the former can produce some dummy activities even if they are not necessary. Therefore, the testing if a given activity network has an event network without dummy activities should be the first step of any method designed to minimize the number of dummy activities, and, as it has been shown, it can be done very efficiently.

4. A GENERAL APPROACH TO THE CONSTRUCTION OF OPTIMAL EVENT NETWORKS

In this and in sections which follow only acircuit digraphs are considered.

The results of the previous section suggest the following general scheme of any algorithm which intends to minimize the number of dummy activities in the event network corresponding to a given activity network D.

INITIALIZATION: Remove all redundant activities from D.

MAIN STEP: If D is a line digraph then there exists an event network of D without dummy activities, otherwise apply an algorithm which minimizes the number of dummy activities. □

There are several algorithms that have so far been proposed and can be incorporated in the main step. We review some of them in the next section and here we present only some basic results which lead to the minimization of the number of nodes in the event networks, since all the algorithms reviewed intend also to minimize that number.

Let $\{ a_i \}$ be the set of activites, and $P(i)$ and $S(i)$ denote respectively the set of immediate predecessors and the set of immediate successors of a_i and $\check{P}(i)$ and $\check{S}(i)$ denote respectively the set of all predecessors and the set of all successors of a_i.

The following lemmas when applied to a set of activities and precedence relations among them produce the event network with the minimum number of nodes (for proofs *see* [2], [3] and [10]).

LEMMA 4.1: *Activities i and j may start at the same node if and only if*

$$P(i) = P(j). \quad \square$$

LEMMA 4.2: *Activities i and j may end at the same node if and only if*

$$S(i) = S(j). \quad \square$$

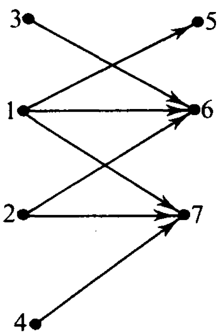
LEMMA 4.3: *The terminal node of activity i may be the initial node of activity j if and only if*

$$\bigcap_{k \in P(j)} \breve{S}(k) = \breve{S}(i), \quad \textit{where} \quad i \in P(j). \quad \square$$

5. A SHORT REVIEW OF ALGORITHMS FOR FINDING AN OPTIMAL EVENT NETWORK

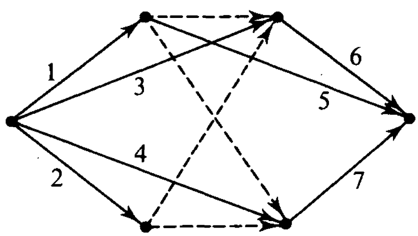
We start this section with an example of an activity network which appears to be very hard for most of the algorithms.

Figure 5.1 (*a*) shows the activity network *D*, and the event network produced by most of the algorithms and the network with the minimum number of dummy activities are shown in figure 5.1 (*b*) and 5.1 (*c*), resp.

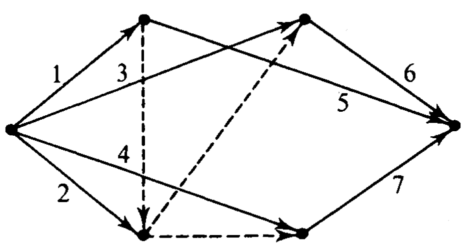


D

(*a*)



(*b*)



E

(*c*)

Figure 5.1.

The first algorithm was proposed by Dimsdal [4] and a counterexample that it does not always produce an event network with the minimum number of nodes and dummy activities is given in [5]. It fails also for the network D of figure 5.1.

The algorithm proposed by Fischer *et al.* [5], contrary to the authors' claim, also fails to create the event network with the minimum number of dummy activities (for instance, for the activity network shown in figure 5.1 (*a*), *see* also [3]). For some activity networks it produces also dummy loops and some parallel dummy activities.

The algorithm presented by Hayes [8] is a set of operations which should be performed to give an event network with the minimum number of dummy activities but, as in the case of two previous algorithms, there is no proof of its correctness and optimality. However it is mentioned in [8] that the number of dummy activities can be decreased by increasing the number of nodes in the event networks.

Cantor and Dimsdal [2] presented the algorithm which for a given digraph constructs the arc-dual digraph with the minimum number nodes but their algorithm introduces redundant dummy activities (*see* the example in [2]) and fails to produce the digraph E for the digraph D of figure 5.1. The algorithm for finding the node-dual digraph with the minimum number of nodes is also presented in [2].

Corneil, Gotlieb and Lee [3] (*see* also [10]) state that if an activity network does not contain parallel activities then the number of nodes and the number of arcs in an event network can be minimized simultanously and that to minimize the latter number we may first minimize the former one and then minimize the latter. Figure 2.3 shows however that in general these statements are not true.

The algorithm of Sterboul and Wertheimer [11] minimizes the number of nodes in the event networks by using the operations which follow from lemmas 4.1-4.3.

6. APPROXIMATE ALGORITHMS WHICH ARE OPTIMAL IN A CERTAIN CLASS OF METHODS

While constructing an event network if we do not intend to minimize neither the number of nodes nor the number of activities then the following event network F can be created immediately. Let D denote an activity network. Then

$$V(F)=\{u_1, u_2 \mid u\in V(D)\} \quad \text{and} \quad A(F)=A_1 \cup A_2,$$

where:

$$A_1 = \{(u_1, u_2) \mid u \in V(D)\} \qquad \text{and} \qquad A_2 = \{(u_2, v_1) \mid (u, v) \in A(D)\}.$$

A_2 is the set of dummy activities.

If D is a digraph then D' is called a *subdivision* of D if it can be obtained from D by a sequence of arc subdivisions. D' is the complete subdivision of D if every arc of D has been subdivided. It is easy to see that $\mathscr{L}(F)$ is isomorphic to the complete subdivision of D and that the dummy activities of F correspond one-to-one with the nodes introduced to D by subdivisions. Evidently, if D is a line digraph then we do not have to subdivide any arc of D to find the digraph F such that $\mathscr{L}(F)=D$. Otherwise, some subdivisions are necessary and the following question arises immediately: for a digraph D, what is the minimum number of arc subdivisions which produce a subdivision D' of D such that D' is a line digraph. This question is answered in [12], where an algorithm for finding D' in polynomial time is also presented. In general, even the minimum number of subdivisions in an activity network produces a great number of dummy activities in the corresponding event network. To improve the method, an *arc set subdivision* has also been defined in [12] which introduces one new node for a subset of arcs which form a complete bipartite subdigraph of D. D' is called a *general subdivision* of D if it can be obtained from D by a sequence of arc set subdivisions. Paper [12] contains a polynomial time algorithm for finding a general subdivision of D which is a line digraph and has the minimum number of new nodes.

It is easy to see that both operations preserve the precedence relations. Once the minimum subdivision or the minimum general subdivision D' of D has been found, the event network F such that $\mathscr{L}(F)=D'$ can easily be constructed. Since both operations: the arc subdivision and the arc set subdivision, and $\mathscr{L}^{-1}$ preserve the precedence relations, the algorithms in [12] produce the approximate solutions to the problem and these solutions are optimal in the classes of all solutions which can by obtained by performing the arc subdivions and the arc set subdivisions, resp.

7. A NEW ALGORITHM FOR FINDING AN OPTIMAL EVENT NETWORK

The approach proposed in this section results from the relations between reversible digraphs and arc-dual digraphs, and leads to the method which can produce the event networks with the minimum number of dummy activities.

We shall make use here of improper covers and improper partitions of a digraph defined in section 2, where we have pointed out that every digraph has an improper cover and a digraph has an improper partition if and only if it is a line digraph.

Notice that in fact every method which for a given activity network constructs an event network E transforms an improper cover of D into an improper partition of a digraph D' such that $V(D) \subseteq V(D')$ and for every pair $u, v \in V(D)$, $u \to v$ in D' if and only if $u \to v$ in D. The elements of the set $V(D') - V(D)$ correspond to the dummy activities of the event network E, and E satisfies $\mathscr{L}(E) = D'$.

Such a transformation consists of a sequence of node insertions which preserve the precedence relations. The nodes inserted correspond to dummy activities of the resulting event network. Now we shall consider the reverse transformation to find a general form of this operation of a node insertion. Suppose that for an acircuit activity network D with no redundant arcs we are given an event network E. If possible, we take E with the minimum number of dummy activities and suppose that E contains some dummy activities. Let $D' = \mathscr{L}(E)$. Vertices in $V(D') - V(D)$ correspond to dummy activities in E. Since D' is a line digraph, it has an improper partition and we take the following one $\langle \{U'_j\}, \{W'_j\} \rangle_J$, where:

$$U'_j = \{(u, v_j) : (u, v_j) \in A(E), u \in V(E)\},$$
$$W'_j = \{(v_j, u) : (v_j, u) \in A(E), u \in V(E)\}$$

and:

$$V(E) = \{v_1, v_2, \ldots, v_m\}, \qquad I = \{1, 2, \ldots, m\}.$$

In other words, a pair $(U'_j;\ W'_j)$ consists of vertices of D' which correspond to arcs coming to and going out of v_j, where $v_j \in V(E)$. Since D is acircuit, so is E and D', we may assume that the pairs $\{(U'_j;\ W'_j)\}_J$ are topologically sorted that is if $v \in W'_i$ and $v \in U'_j$ then $i < j$. The following algorithm transforms the improper partition $\langle \{U'_j\}, \{W'_j\} \rangle_J$ of D' into an improper cover of D.

ALGORITHM EA (from Event network to Activity network):

1. Set $W_j = W'_j$, $j = 1, 2, \ldots, m$.
2. Find maximal index i_0 such that W_{i_0} contains a dummy arc v. If no such i_0 exists, then go to 6.
3. Suppose v is in U'_{j_0} $(j_0 > i_0)$. Replace W_{i_0} by $W_{i_0} - \{v\} \cup W_{j_0}$.
4. Scan if the redundancy has been introduced by step 3. If there exist $v \in U'_{i_0}$, $u, w \in W_{i_0}$ and index k_0 such that $u \in U'_{k_0}$ and $w \in W_{k_0}$ then remove w from W_{i_0}.
5. Go to 2.

6. Remove all dummy arcs from the sets $\{U_i'\}$, $U_i = U_i'$ and remove all pairs $(U_j; W_j)$ such that $U_j = \emptyset$ except the pair for which $U_j = \emptyset$, $W_j \neq \emptyset$ and $\Gamma_D^{-1} W_j = \emptyset$. Let $\langle \{U_i\}, \{W_i\} \rangle_I$, where $I = \{1, 2, \ldots, n\}$ denotes the created family of sets. □

The following properties of $\langle \{U_i\}, \{W_i\} \rangle_I$ follows from the relations between digraph D and digraphs E and D', and from Algorithm EA.

PROPERTY 7.1: *The pairs* $\{(U_i; W_i)\}_I$ *are topologically sorted.* □

PROPERTY 7.2: *For every maximal set* $Q = \{u, v \in V(D) : \Gamma_D u = \Gamma_D v\}$ *and* $R = \{u, v \in V(D) : \Gamma_D^{-1} u = \Gamma_D^{-1} v\}$ *there exist* k, $l \in I$ *such that* $Q = U_k$ *and* $R \subseteqq W_l$, *resp.* □

PROPERTY 7.3: *The families* $\{U_i\}_I$ *and* $\{W_i\}_I$ *constitue an improper partition and an improper cover of* $V(D)$, *resp., and* $A(D) = \bigcup_{i \in I} U_i \times W_i$. □

Now we are ready to present an algorithm which for a given improper cover of an activity network D creates an improper partition of a line digraph D' and tends to minimize the number $V(D') - V(D)$. The digraph (or multidigraph) E which satisfies $\mathscr{L}(E) = D'$ is the event network corresponding to the activity network D.

Let D be an activity network. The algorithm starts with the improper cover $\langle \{U_i\}, \{W_i\} \rangle_I$ of D, where U_i is the maximal set of vertices with the some successors and $W_i = \Gamma_D U_i$, $i \in I = \{1, 2, \ldots, n\}$. The families $\{U_i\}_I$ and $\{W_i\}_I$ satisfy property 7.3 and the algorithm transforms them into two improper partitions of a set which contains $V(D)$. The transformation consists of a sequence of node insertions which are created when some nodes of D occure in more than one set W_i. By lemma 4.1, the activities with the same set of predecessors appear in the same set W_j and they may be considered together in the algorithm and by lemma 4.3, an activity $j \in W_k$ for which there exists an activity $i \in U_k$ such that i and j satisfy the condition of the lemma need not be moved from W_k, such an activity j is said to be *stable* in W_k.

The algorithm works in the direction opposite to that of Algorithm EA, therefore we assume that (1) the pairs $\{(U_i; W_i)\}_I$ are topologicaly sorted, i. e., if $u \in W_i$ and $u \in U_j$ then $i < j$; (2) let $u \in W_j$ be a stable activity in W_j. If there exists $i \neq j$ such that $u \in W_i$ then $i < j$; and (3) if $W_j \subset W_i$ then $i < j$. Notice that rules (1)-(3) do not order $\{(U_i; W_i)\}_I$ uniquely.

ALGORITHM AE (from Activity network to Event network).

Initialization : Let D be an acircuit digraph without redundant arcs.

1. Form two families $\{U_i\}_I$ and $\{W_i\}_I$ of subsets of $V(D)$ such that U_i is the maximal set of nodes with the same set of successors and $W_i = \Gamma_D U_i$ $(i \in I)$. If $\{W_i\}_I$ is an improper partition of $V(D)$ then go to 9.

2. Order the pairs $\{(U_i;\ W_i)\}_I$ according to the rules (1)-(3) given above. Set $h=0$.

Node insertions:

3. Find minimal index i_0 such that W_{i_0} contains a node which belongs to another set $W_j (j > i_0)$. If no such i_0 exists then go to 9.

4. Find a minimal number of sets $W_{k_1}, W_{k_2}, \ldots, W_{k_p} (k_j > i_0)$ which either cover non-stable activities of W_{i_0} or cover non-stable activities of W_{i_0} and some activities which are immediate successors of activities in W_{i_0}. If W_{i_0} or a part of it cannot by covered in such a way then go to 7.

5. Introduce new nodes $z_{h+1}, z_{h+2}, \ldots, z_{h+p}$ as follows. Replace non-stable covered nodes of W_{i_0} by $z_{h+1}, z_{h+2}, \ldots, z_{h+p}$ and set $U_{k_j} = U_{k_j} \cup \{z_{h+j}\}$ for $j = 1, 2, \ldots, p$. Set $h = h+p$.

6. Go to 3.

7. Find a maximal subset Y of activities which are non-stable in sets W_{i_1}, $W_{i_2}, \ldots, W_{i_q}$ and q is maximal. Introduce new nodes $z_{h+1}, z_{h+2}, \ldots, z_{h+q}$ and a new pair $(z_{h+1}, z_{h+2}, \ldots, z_{h+q};\ Y)$ as follows. Replace Y by z_{h+j} in $W_{i_j} (j = 1, 2, \ldots, q)$ and locate the new pair immediately after the i_s-th pair, where $i_s = \max\{i_j : j = 1, 2, \ldots, q\}$. Set $h = h+q$.

8. Go to 3.

Termination:

9. Let $\{(U'_i;\ W'_i)\}_J$ denote the family of pairs created in step Node Insertions. □

It is easy to prove the following properties.

PROPERTY 7.4: *If* $\{(U'_i;\ W'_i)\}_J$ *is the family of pairs created by algorithm* AE, *then:* (i) $\bigcup_{i \in J} U'_i = \bigcup_{i \in J} W'_i$; (ii) $\{U'_i\}_J$ *and* $\{W'_i\}\}_J$ *are improper partitions of* $X = \bigcup_{i \in J} U'_i$, (iii) *if* D' *denotes the digraph for which* $V(D') = X$ *and* $\langle \{U'_i\}, \{W'_i\} \rangle_J$ *is an improper partition, then the digraph* E *such that* $\mathscr{L}(E) = D'$ *is an event network of* D; (iv) E *is acircuit, and* (v) E *contains no redundant arcs.* □

The last property 7.4 (v) follows from the fact that D has no redundant arcs and that in step 4 we cover activities which are in distance of at most 2 from activities in U_{i_0}.

In general, Algorithm AE produces only a suboptimal solution (*see* examples 7.4 and 7.5) however one can easily verify that it needs only polynomial time.

Since the problem of finding an event network with the minimum number of dummy activities is NP-complete, it is unlikely that any polynomial time algorithm can produce an optimal solution for every input. However it is not difficult to show how using slightly modified versions of steps 4-5 and 7 to design a branch and bound algorithm which will always solve the problem. To this end we should take into consideration that the initial order of pairs $\{(U_i;\ W_i)\}_I$ is not unique and that the result of the algorithm depends on the order of steps 4 and 7 in the sequence of node insertions. The details are left to the Reader since our goul was only to present an approach which can lead to the optimal solution and to give an approximate algorithm, and the generalization is straigthforward. We conclude this section with a number of examples which illustrate the main steps and features of Algorithm AE.

Example 7.1: Let consider the activity network D shown in figure 5.1 (*a*).

TABLE 7.1.

i	U_i	W_i	i	U_i	W_i	i	U_i	W_i
1...	$\emptyset$	1, 2, 3, 4	1...	$\emptyset$	1, 2, 3, 4	1..	$\emptyset$	1, 2, 3, 4
2...	1	$\underline{5}$, 6, 7	2...	1	$\underline{5}$, z_1	2..	1	5, z_1
3...	2	6, 7	3...	2, z_1	6, 7	3..	2, z_1	z_2, z_3
4...	3	$\underline{6}$	4...	3	$\underline{6}$	4..	3, z_2	6
5...	4	$\underline{7}$	5...	4	$\underline{7}$	5..	4, z_3	7
6...	5, 6, 7	$\emptyset$	6...	5, 6, 7	$\emptyset$	6..	5, 6, 7	$\emptyset$

The first three columns of table 7.1 show the families $\{U_i\}$ and $\{W_i\}$ after two steps of Initialization. The stable activities are underlined. First, the algorithm (step 3) finds $i_0 = 2$ and two non-stable activities 6 and 7 which can be covered by W_3. Columns 4-6 of table 7.1 show the families $\{U_i\}$ and $\{W_i\}$ after the insertion of node z_1. In the next interation (step 3), W_3 can be covered by W_4 and W_5. The last three columns show the improper cover of a digraph D' which corresponds to the event network with the minimum number of dummy activities. □

Example 7.2: Table 7.2 in the first three columns contains ordered pairs of subsets of the improper cover of a network and the last three columns contain the solution obtained by applying step 7 of the algorithm.

TABLE 7.2

i	U_i	W_i	i	U_i	W_i
1.	$\emptyset$	a, d	1.	$\emptyset$	a, d
2.	b	$\underline{e}, f, g$	2.	b	e, z_1
3.	c, d	$f, g, \underline{h}$	3.	c, d	z_2, h
4.	e, f, g, h	$\emptyset$	4.	z_1, z_2	f, g
			5.	e, f, g, h	$\emptyset$

Example 7.3: Let now consider the activity network D which corresponds to the node-cover problem for the graph shown in figure 7.1 (*see* also [9]).

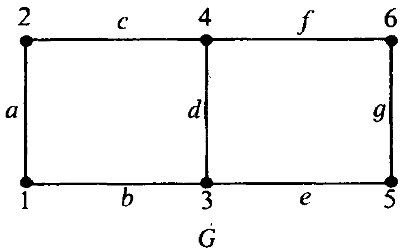


Figure 7.1.

The set of activities of D consists of the nodes and the edges of G and one additional activity x. Precedence relations of D are as follows:

$$v<u \text{ if either edge } v \text{ is incident with node } u$$
$$\text{or } v \text{ is an edge and } u=x,\ v,\ u\in V(D).$$

Paper [9] contains the proof that the minimum node-cover problem in graphs with vertices of degree two or three, which is NP-complete, is polynomially transformable to the problem of finding the event network with the minimum number of dummy activities for the activity network constructed above. Let apply Algorithm AE to the activity network corresponding to the graph G of figure 7.1.

TABLE 7.3.

i	U_i	W_i	i	U_i	W_i	i	U_i	W_i
1...	$\emptyset$	$a, b, \ldots, g$	1..	$\emptyset$	$a, b, \ldots, g$	1..	$\emptyset$	$a, b, \ldots, g$
2...	a	$1, 2, x$	2..	a	$1, z_4$	2..	a	z_4, z_{14}
3...	b	$1, 3, x$	3..	b	$1, z_1$	3..	b	z_1, z_{15}
4...	c	$2, 4, x$	4..	c	$4, z_5$	4..	z_{14}, z_{15}	1
5...	d	$3, 4, x$	5..	z_4, z_5	$2, x$	5..	c	z_5, z_8
6...	e	$3, 5, x$	6..	d	$4, z_2$	6..	z_4, z_5	$2, z_{11}$
7...	f	$4, 6, x$	7..	e	$5, z_3$	7..	d	z_2, z_9
8...	g	$5, 6, x$	8..	z_1, z_2, z_3	$3, x$	8..	e	z_3, z_{16}
9...	$1, 2, \ldots, 6, x$	$\emptyset$	9..	f	$4, z_6$	9..	z_1, z_2, z_3	$3, z_{12}$
			10..	g	$5, z_7$	10..	f	z_6, z_{10}
			11..	z_6, z_7	$6, x$	11..	z_8, z_9, z_{10}	4
			12..	$1, 2, \ldots, 6, x$	$\emptyset$	12..	g	z_7, z_{17}
						13..	z_{16}, z_{17}	5
						14..	z_6, z_7	$6, z_{13}$
						15..	z_{11}, z_{12}, z_{13}	x
						16..	$1, 2, \ldots, 6, x$	$\emptyset$

The first three columns of table 7.3 contain the improper cover of D after Initialization. Step 4 cannot be performed for any W_i, and in step 7 we first find $Y=\{3, x\}$ which belongs to W_3, W_5 and W_6, and then $Y=\{2, x\}$ and $Y=\{6, x\}$ are found. Columns 4-6 show the families $\{U_i\}$ and $\{W_i\}$ after performing these three node insertions. In the next steps we have $Y=\{4\}$, $Y=\{x\}$, $Y=\{1\}$ and $Y=\{5\}$, and finally we obtain the improper cover of D' which is shown in the last three columns of table 7.3. The node-cover corresponding to the event network E such that $\mathscr{L}(E)=D'$ consists of three vertices $\{2, 3, 6\}$. It is a minimum cover and therefore, by the results of [9], E has the minimum number of dummy activities. □

Example 7.4: Consider now the activity network D shown in figure 2.3 (*a*) and its improper cover ($\emptyset$; a, b, b', c, g, k, l), (a; **h**, g), (b; d, e, f, **i**), (b'; d, e, f, **m**), (c; d, f, j), (g; **d**), (k; **e**), (l; **f**), and (d, e, f, h, i, j, m; $\emptyset$). Following precisely the steps of algorithm AE, we obtain the event network E shown in figure 2.3 (*b*). Suppose however that we perform step 4 for activity e in W_2, step 7 for $Y=\{d, e, f\}$ in W_3 and W_4, and then step 4 for successive sets W_i if necessary. Finally we obtain the event network with the minimum number of dummy activities described at the end of section 2. One can easily verify that if step 7 is all the time performed before step 4 then we get the event network which has also the minimum number of dummy activities however it has one more node then the previous one. □

Example 7.5: Let an activity network be given by its improper cover shown in the first three columns of table 7.4. W_2 can be covered by W_3 and W_4. W_4 is contained in $W_5 \cup W_6$ and $g \in (W_5 \cup W_6) - W_4$ is an immediate successor of $k \in W_4$ therefore W_4 can also be covered by W_5 and W_6. Columns 4-6 of table 7.4 show the improper cover after these two steps. Next, activity g in W_6 can be covered by W_7 because activities w and m are the immediate successors of s. The

TABLE 7.4.

i	U_i	W_i	i	U_i	W_i	i	U_i	W_i
1. . . .	$\emptyset$	c, d, h, p, v	1. . .	$\emptyset$	c, d, h, p, v	1. . .	$\emptyset$	c, d, h, p, v
2. . . .	c	r, q, k, s	2. . .	c	z_1, z_2	2. . .	c	z_1, z_2
3. . . .	d	$\underline{r}, s$	3. . .	z_1, d	$\underline{r}, s$	3. . .	z_1, d	r, z_8
4. . . .	p	q, k, s	4. . .	z_2, p	z_3, z_4	4. . .	z_2, p	z_3, z_4
5. . . .	v	$\underline{k}, s$	5. . .	z_3, v	$\underline{k}, s$	5. . .	z_3, v	k, z_9
6. . . .	h	$\underline{q}, g, s$	6. . .	z_4, h	$\underline{q}, g, s$	6. . .	z_4, h	$q; z_5, z_{10}$
7. . . .	k	$\underline{g}, w, m$	7. . .	k	$\underline{g}, w, m$	7. . .	z_8, z_9, z_{10}	s
8. . . .	s	$\underline{w}, m$	8. . .	s	$\underline{w}, m$	8. . .	z_5, k	g, z_6
9. . . .	q	$\underline{m}$	9. . .	q	$\underline{m}$	9. . .	z_6, s	w, z_7
10. . . .	g, m, r, w	$\emptyset$	10. . .	g, m, r, w	$\emptyset$	10. . .	z_7, q	m
						11. . .	g, m, r, w	$\emptyset$

non-stable activities w and n in W_7 can be covered by W_8, and m in W_8 — by W_9. Finally we apply step 7 to $Y = \{ s \}$. The last three columns in Table 7.4 show the solution obtained by Algorithm AE.

ACKNOWLEDGEMENTS

The author is grateful to Professor Masao Iri from University of Tokyo and to Professor Narsingh Deo from Washington State University for discussion and helpful comments, and to the referees for substantive suggestions.

REFERENCES

1. A. V. AHO, M. R. GAREY and J. D. ULLMAN, *The Transitive Reduction of a Directed Graph*, S.I.A.M. J. Comput., Vol. 1, 1972, pp. 131-137.
2. D. G. CANTOR and B. DIMSDALE, *On Direction-Preserving Maps of Graphs*, J. Combinatorial Theory, Vol. 6, 1969, pp. 165-176.
3. D. G. CORNEIL, C. C. GOTLIEB and Y. M. LEE, *Minimal Event-Node Network of Project Precedence Relations*, Comm. A.C.M., Vol. 16, 1973, pp. 296-298.
4. B. DIMSDALE, *Computer Construction of Minimal Project Networks*, I.B.M. Syst. J., Vol. 2, 1963, pp. 24-36.
5. A. C. FISHER, J. S. LIEBMAN and G. L. NEMHAUSER, *Computer Construction of Project Networks*, Comm. A.C.M., Vol. 11, 1968, pp. 493-497.
6. F. HARARY and Z. R. NORMAN, *Some Properties of Line Digraphs*, Rend. Circ. Mat. Palermo, Vol. 9, 1960, pp. 161-168.
7. F. HARARY, *Graph Theory*, Addison-Wesley, Mass., 1969.
8. M. HAYES, *The Role of Activity Precedence Relationships in Node-Oriented Networks*, in: Project Planning by Network Analysis, Proccedings of the Second International Congress, North-Holland Pub. Comp., Amterdam, 1969, pp. 128-146.
9. M. S. KRISHNAMOORTHY and N. DEO, *Complexity of the Minimum-Dummy-Activities Problem in a PERT Network*, Networks, Vol. 9, 1969, pp. 189-194.
10. Y. M. LEE, *The Minimal Event-Node Network of Project Precedence Relations*, M. Sc. Thesis, University of Toronto, Canada, 1971.
11. F. STERBOUL and D. WERTHEIMER, *Comment construire un graphe PERT minimal*, Université des Sciences et Techniques de Lille, January 1979.
12. M. M. SYSLO, *Remarks on Line Digraphs*, Bull. Acad. Polon. Sc., Ser. Sc. Math. Astronom. Phys., Vol. 22, 1974, pp. 5-10.
13. M. M. SYSLO, *Optimal Constructions of Reversible digraphs*, Technical Report Nr. N-55, Institute of Computer Science University of Wroclaw, Wroclaw, May 1979.