RAIRO. RECHERCHE OPÉRATIONNELLE

DAVID J. MICHAEL
JERZY KAMBUROWSKI

On the minimum dummy-arc problem

RAIRO. Recherche opérationnelle, tome 27, nº 2 (1993), p. 153-168
<http://www.numdam.org/item?id=RO_1993__27_2_153_0>

Recherche opérationnelle/Operations Research
(vol. 27, n° 2, 1993, p. 153 à 168)

ON THE MINIMUM DUMMY-ARC PROBLEM (*)

by David J. MICHAEL (1), Jerzy KAMBUROWSKI (2)
and Matthias STALLMANN (3)

Abstract. – *A precedence relation can be represented non-uniquely by an activity on arc (AoA) directed acyclic graph (dag). This paper deals with the NP-hard problem of constructing an AoA dag having the minimum number of arcs among those that have the minimum number of nodes. We show how this problem can be reduced in polynomial time to the set-cover problem so that the known methods of solving the set-cover problem can be applied. Several special cases that lead to easy set-cover problems are discussed.*

Keywords: Activity networks, dummy activities, set-cover.

Résumé. – *Une relation de précédence peut être représentée (d'une manière non-unique) par un graphe direct acyclique (gda) avec activités sur arcs (AsA). Cet article traite du problème NP-difficile de la construction d'un gda AsA ayant le nombre minimal d'arcs parmi ceux qui ont le nombre minimal de sommets. Nous montrons comment ce problème peut se réduire en un temps polynomial au problème de recouvrement d'ensembles, en sorte que l'on peut appliquer les méthodes connues de résolution de ce dernier problème. Nous examinons plusieurs cas spéciaux qui conduisent à des problèmes faciles de recouvrements d'ensembles.*

Mots clés : Réseaux d'activités, activités fictives, recouvrement d'ensembles.

1. INTRODUCTION

The activities of a project are often constrained by conditions such as "activity v cannot start until activity u has finished". Assuming that no activity is repeated we can define a precedence relation $\prec$ on the activities, so that $u \prec v$ means that u must finish before v starts. The relation $\prec$ can be represented graphically in two different ways, by either assigning the

(*) Received November 1990.

(1) International Business Machines Corporation, P.O. Box 60000, Cary, NC 27511
(2) Department of Information Systems and Operations Management, University of Toledo, Toledo, OH 43606-3390.
(3) Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206.

Recherche opérationnelle/Operations Research, 0399–0559/93/02 153 16/$3.60

activities to the nodes or to (a subset of) the arcs. In either case a *directed acyclic graph (dag)* is defined. In an *activity on node* (AoN) dag each activity corresponds one-to-one with a node, and we say that $u \prec v$ is *represented* if there is a directed path of arcs leading from v's node to u's node. Thus an AoN dag is unique except for possible transitive arcs. In an *activity on arc* (AoA) dag, each activity v corresponds to an arc, where parallel arcs that share the same start and terminal nodes are permitted. We say $u \prec v$ is *represented* in an AoA dag if there is a path from t_u, the terminal node of the arc for u, to s_v, the start node of the arc for v (the path is empty if $t_u = s_v$). Additional *dummy arcs* may have to be added to represent all constraints of $\prec$, and no canonical method for adding dummy arcs has been agreed to. Thus an AoA dag is not unique.

This paper shows how to construct an AoA dag that has the minimum number of dummy arcs given that it has the minimum number of nodes. This defines what we will refer to here as the *dummy-arc* problem. Note that this definition implies that AoA dags have one initial node and one terminal node. Sysło [14] gives a good overview of the problem and provides a simple counter-example that shows we cannot minimize both the number of arcs and the number of nodes simultaneously. The problem of minimizing only the number of nodes can be solved in polynomial time using the algorithm of Cantor and Dimsdale [1], or the algorithm of Sterboul and Wertheimer [12].

The dummy-arc problem was shown by Krishnamoorthy and Deo [7] to be NP-hard. Several heuristics have been proposed, some of which construct an AoA dag directly, while others construct a dual graph. These include algorithms proposed by Corneil *et al.* [3], Dimsdale [4], Fisher *et al.* [5], Hayes [6], Spinrad [13], and Sysło [14, 15, 16]. Mrozek [10] gives an algorithm to verify if heuristically produced solutions are optimal. Only Corneil *et al.* claimed to have an optimal algorithm, but this was disproved by Mrozek [11]. Some of the heuristics will solve the problem for very restricted classes of precedence relations.

We solve the dummy-arc problem by showing how an instance of the dummy-arc problem may be reduced (in polynomial time) to an instance of the well-known set-cover problem. This allows us to solve the dummy-arc problem, either heuristically or optimally, using established set-cover algorithms and heuristics. Such algorithms are reviewed by Christofides and Korman [2], while heuristics are reviewed by Vasko and Wilson [17]. Mrozek [11] gives nearly the same reduction to the set-cover problem. However, our reduction is much simpler in presentation, it leads to a more concise instance of the set-cover problem, and we show that it subsumes efficient algorithms for several previously studied and some new special cases.

After developing some simple notation at the end of this section, we present a simple construction of the minimum set of nodes for AoA dags in section 2, and develop the reduction in section 3. Section 4 describes special cases for which the dummy-arc problem can be solved in polynomial time.

We let G refer to the AoN dag of a given precedence relation. We assume that G is transitively reduced, *i.e.*, $(u,v) \in G$ implies there exists no activity (node) w such that $(u,w) \in G$ and $(w,v) \in G$. The transitive closure of G is denoted by $tc(G)$, where $(u,v) \in tc(G)$ if there is a (possibly empty) path in G from node u to node v. Let $P(v)$ and $S(v)$ denote the sets of immediate predecessors and successors of activity v:

$$P(v) = \{u \mid (u, v) \in G\},$$
$$S(v) = \{w \mid (v, w) \in G\}.$$

Let $P^*(v)$ and $S^*(v)$ be the sets of all (not necessarily immediate) predecessors and successors of v:

$$P^*(v) = \{u \mid (u, v) \in tc(G)\},$$
$$S^*(v) = \{w \mid (v, w) \in tc(G)\}.$$

Note that $P^*(u) \subseteq P^*(v)$ iff $S^*(v) \subseteq S^*(u)$, but that this is not necessarily true for $P(v)$ and $S(v)$. We extend our terminology to say that a *constraint* $(u,v) \in G$ is *represented* in an AoA dag D if there is a path in D from t_u to s_v.

2. CONSTRUCTION OF THE MINIMUM SET OF NODES

The construction of the minimum set of nodes was first developed by Cantor and Dimsdale [1]. Their construction can be simplified as follows. Another version of this algorithm can be found in Sysło [16]. Recall that (s_v, t_v) is the arc of an AoA dag corresponding to activity $v \in G$. For each $v \in G$, we define two pairs of activity-sets, denoted $(P^*(s_v), S^*(s_v))$ and $(P^*(t_v), S^*(t_v))$, where

$$P^*(s_v) = P^*(v) \quad \text{and} \quad S^*(s_v) = \bigcap_{u \in P(v)} S^*(u),$$
$$S^*(t_v) = S^*(v) \quad \text{and} \quad P^*(t_v) = \bigcap_{w \in S(v)} P^*(w),$$

provided that the intersection over an empty set is equal to the set of all activities.

The minimum set of nodes is defined by the set of distinct pairs of activity-sets. There is then a single activity-set pair, denoted $(P^*(j), S^*(j))$, for each node j. The set $P^*(j)$ is the set of activities that precede node j, while $S^*(j)$ is the set of activities that follow node j. The construction of the minimum set of nodes is illustrated using the dag G shown in Figure 1. Table I lists the pairs of activity-sets for each of the ten activities, while Table II lists the nine distinct pairs defining the minimum set of nodes. By construction we have $s_v = i$ if $P^*(i) = P^*(s_v)$, and $t_v = j$ if $S^*(j)=S^*(t_v)$. The (AoA) *framework*, depicted by the solid arcs in Figure 2, is the set of activity arcs on the minimum set of nodes, but does not include any dummy arcs. In general, the framework may have many initial nodes and many terminal nodes. However, it always has a single initial node $s_\emptyset$ with $P^*(s_\emptyset) = \emptyset$, and a single terminal node $t_\emptyset$ with $S^*(t_\emptyset) = \emptyset$. The nodes $s_\emptyset$ and $t_\emptyset$ represent the project initiation and the project termination in every AoA dag having the minimum number of nodes. In Figure 2, $s_\emptyset = 1$ and $t_\emptyset = 9$.

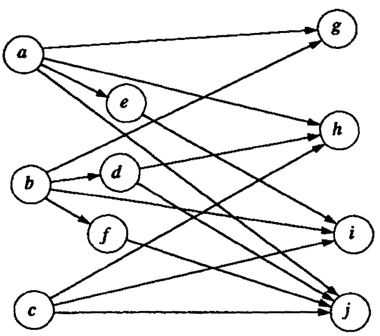

Figure 1

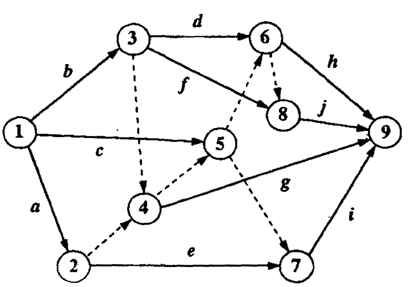

Figure 2

3. REDUCTION TO THE SET-COVER PROBLEM

A constraint $(u,v) \in G$ is not represented in the framework defined by G if $t_u \neq s_v$. In order to represent (u,v), a dummy path $\pi(t_u, s_v)$ (a path of dummy arcs which leads from t_u to s_v) must be added to the framework. Identifying all unrepresented constraints can be inefficient because two or more activities may share the same start and/or finish nodes. Therefore, we will instead identify the set R of all node pairs (i, j) of the framework that must be represented (*i.e.*, connected) by dummy paths,

$$R = \{(i, j) \mid i \neq j \text{ and } \exists (u, v) \in G \text{ such that } t_u = i \text{ and } s_v = j\}.$$

TABLE I

Activity-set-pairs

v	$P^*(s_v)$	$S^*(s_v)$	$P^*(t_v)$	$PS^*(t_v)$
a	–	*a b c d e f g h i j*	*a*	*e g h i j*
b	–	*a b c d e f g h i j*	*b*	*d f g h i j*
c	–	*a b c d e f g h i j*	*a b c*	*h i j*
d	*b*	*d f g h i j*	*a b c d*	*h j*
e	*a*	*e g h i j*	*a b c e*	*i*
f	*b*	*d f g h i j*	*a b c d f*	*j*
g	*a b*	*g h i j*	*a b c d e f g h i j*	–
h	*a b c d*	*h j*	*a b c d e f g h i j*	–
i	*a b c e*	*i*	*a b c d e f g h i j*	–
j	*a b c d f*	*j*	*a b c d e f g h i j*	–

TABLE II

Minimum Set of Nodes

i	$P^*(i)$	$S^*(i)$	*Labels*
1	–	*a b c d e f g h i j*	$s_a\ s_b\ s_c$
2	*a*	*e g h i j*	$t_a\ s_e$
3	*b*	*d f g h i j*	$t_b\ s_d\ s_f$
4	*a b*	*g h i j*	s_g
5	*a b c*	*h i j*	t_c
6	*a b c d*	*h j*	$t_d\ s_h$
7	*a b c d e*	*i*	$t_e\ s_i$
8	*a b c d f*	*j*	$t_f\ s_j$
9	*a b c d e f g h i j*	–	$t_g\ t_h\ t_i\ t_j$

A set of dummy arcs D is a (feasible) *solution* to the dummy-arc problem if D represents all pairs in R, *i.e.*, if there exists a dummy path $\pi(i, j)$ in D for every $(i, j) \in R$. An optimal solution is a solution D for which $|D|$ is minimum. A pair (i, j) is *feasible* if $P^*(i) \subset P^*(j)$ (equivalently $S^*(j) \subset S^*(i)$). In other words, a pair is feasible if it can be added as a dummy arc without introducing any constraints not consistent with $\prec$. This definition of feasible allows redundant pairs (i, j) for which there is a path in the framework from i to j, but it is evident from subsequent definitions that such pairs are not used in our reduction.

For each pair $(i, j) \in R$ we define

$$X(i, j) = \{k \mid (k, j) \in R \text{ and } (i, k) \text{ is feasible}\},$$
$$Y(i, j) = \{l \mid (i, l) \in R \text{ and } (l, j) \text{ is feasible}\}.$$

If $(i, j) \in R$ and both (i, k) and (k, j) are feasible, then $k = t_x$ for some activity x implies $k \in X(i, j)$, and $l = s_y$ for some activity y implies $l \in Y(i, j)$. Therefore, each dummy path $\pi(i, j)$ representing $(i, j) \in R$ can cross only the nodes of $X(i, j) \cup Y(i, j)$. The nodes of $Y(i, j)–X(i, j)$ and $X(i, j)–Y(i, j)$ are initial and terminal nodes in the framework, respectively.

The set R may be divided into three mutually exclusive subsets as follows:

$$R_1 = \{(i, j) \mid X(i, j) \cap Y(i, j) \neq \emptyset\},$$
$$R_2 = \{(i, j) \mid X(i, j) \cup Y(i, j) = \emptyset\},$$
$$R_3 = R - (R_1 \cup R_2).$$

For $k \in X(i, j) \cap Y(i, j)$, both (i, k) and (k, j) belong to R. Therefore, the pairs (i, k) and (k, j) must be represented by dummy paths $\pi(i, k)$ and $\pi(k, j)$. Since the path $\pi(i, j) = \pi(i, k)\ \pi(k, j)$ represents (i, j), we can say that R_1 is the subset of those pairs in R that are "automatically" represented. When $X(i, j) \cup Y(i, j) = \emptyset$, the pair (i, j) can only be represented by the dummy arc (i, j) and R_2 contains such pairs. The set R_3 contains all pairs not in R_1 or R_2, and it is these pairs that make the dummy-arc problem intractable. In Figure 2 we have $R_1 = \{(2,8), (5,8)\}$, $R_2 = \{(2,4), (3,4), (5,6), (5,7), (6,8)\}$, and $R_3 = \{(2,6), (3,7)\}$.

For each $(i, j) \in R_3$ we define the following set of dummy arcs:

$$F(i, j) = F_1(i, j) \cup F_2(i, j) \cup F_3(i, j),$$

where

$$F_1(i, j) = \{(i, k) \mid k \in X(i, j)\},$$
$$F_2(i, j) = \{(l, j) \mid l \in Y(i, j)\},$$
$$F_3(i, j) = \{(l, k) \mid (l, k) \text{ is feasible, } k \in X(i, j),\ l \in Y(i, j)\}$$

For example, for $(2,6) \in R_3$ we have $X(2,6) = \{5\}$ and $Y(2,6) = \{4\}$, so $F_1(2,6) = \{(2,5)\}$, $F_2(2,6) = \{(4,6)\}$, $F_3(2,6) = \{(4,5)\}$ and $F(2,6) = \{(2,5), (4,5), (4,6)\}$.

The set $F(i, j)$ consists of dummy arcs associated with the pair $(i, j) \in R_3$ that are sufficient to represent this pair, under the assumption that the remaining pairs are already represented. Figure 3 show how the three types of dummy arcs in $F(i, j)$ (shown as dashed lines) interact with paths representing other pairs in R (shown as wavy lines). For example, if $k \in X(i, j)$ then $(k, j) \in R$

and assuming that (k, j) is already represented, it suffices to add (i, k) to represent (i, j). Note that the dummy arc (i, j) is not included in $F(i, j)$, although there may exist optimal solutions to which this arc belongs. Leaving (i, j) out of $F(i, j)$ allows us to reduce the dummy-arc problem to a more concise instance of the set-cover problem, and the following lemma validates this approach.

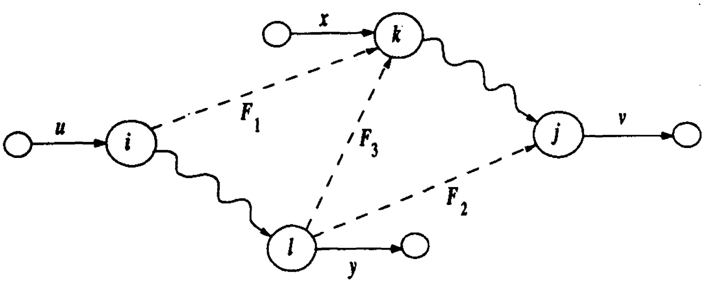

Figure 3

LEMMA 1: *If D is a solution to the dummy-arc problem, there exists a solution D' with $|D'| \leq |D|$ and $D' \cap R_3 = \varnothing$.*

Proof: Suppose $D \cap R_3 \neq \varnothing$ for a solution D. It suffices to show that there exists a solution D'' such that $|D''| \leq |D|$ and $|D'' \cap R_3| < |D \cap R_3|$. Let $(i, j) \in D \cap R_3$. This implies

$$X(i, j) \cap Y(i, j) = \varnothing \quad \text{and} \quad X(i, j) \cup Y(i, j) \neq \varnothing.$$

If $k \in X(i, j)$, the dummy arc $(i, k) \notin R_3$ because otherwise $k \in Y(i, j)$. Letting $D'' = D - \{(i, j)\} \cup \{(i, k)\}$ we have

$$|D''| \leq |D| \quad \text{and} \quad |D'' \cap R_3| < |D \cap R_3|.$$

To see that D'' is a solution, we need only verify that the pair (i, j) is represented by D''. Since $(k, j) \in R$, there must exist a dummy path $\pi(k, j)$ in D. This path is also in D'' and together with arc (i, k) represents (i, j) in D''

A symmetric argument can be made if $l \in Y(i, j)$, and then we let $D'' = D - \{(i, j)\} \cup \{(l, j)\}$. ■

Recall that an arbitrary instance of the set-cover problem is defined by a collection C of subsets of a finite set Q. A solution is a subset $C' \subseteq C$ such that every element of Q belongs to at least one member of C'. The subset C' is *optimal* if C' is a solution and $|C'| \leq |C''|$ for all solutions $C'' \subseteq C$.

We define an instance of the set-cover problem as follows:

$$Q = R_3,$$
$$Q_{pq} = \{(i, j) \in R_3 \mid (p, q) \in F(i, j)\},$$
$$C = \{Q_{pq} \mid (p, q) \in F\},$$

where

$$F = \bigcup_{(i, j) \in R_3} F(i, j)$$

In other words, the elements of Q are the pairs of R_3. If a pair $(i, j) \in R_3$, there is a set $Q_{pq} \in C$ for each dummy arc $(p, q) \in F(i, j)$. The set Q_{pq} contains all pairs of R_3 with which the dummy arc (p, q) is associated. The elements of F are those dummy arcs that can be useful in representing the pairs of R_3, and $|Q_{pq}| \geq 1$ for every $(p, q) \in F$. Mrozek's reduction [11] includes sets for which $Q_{pq} = \emptyset$. In the example of Figures 1 and 2, dummy arc (4,8) with $Q_{48} = \emptyset$ is considered by Mrozek's reduction but not by ours.

A solution of size K to this set-cover problem is a subset $C' \subseteq C$ such that $|C'| = K$, and for every pair $(i, j) \in R_3$ there exists $Q_{pq} \in C'$ which contains the pair.

THEOREM 1: *The set-cover problem has a solution C' with $|C'| \leq K$ if and only if the dummy-arc problem has a solution D' with $|D'| \leq K + |R_2|$.*

Proof: Suppose C' is a solution to the set-cover problem with $|C'| \leq K$. We show there exists a corresponding set of $|C'| + |R_2|$ dummy arcs, $D' = \{(p, q) \mid Q_{pq} \in C'\} \cup R_2$, such that every $(i, j) \in R$ is represented. We use induction on the following containment relation among the pairs in R:

$$(i, j) \in R \, contains \, (k, l) \in R \quad \text{if} \quad (k, l) \neq (i, j),$$
$$S^*(k) \subseteq S^*(i) \quad \text{and} \quad P^*(l) \subseteq P^*(j).$$

Note that if $k \in X(i, j)$, then (i, j) contains (k, j), and if $l \in Y(i, j)$, then (i, j) contains (i, l). Note also that the containment relation defines a strict partial order on R; hence the induction is valid.

For the basis case we consider all pairs (i, j) that contain no other pairs, so $X(i, j) \cup Y(i, j) = \emptyset$. By definition, $(i, j) \in R_2$ and (i, j) is represented by the dummy arc (i, j).

For the induction step we now suppose that all pairs contained by (i, j) are represented. Suppose there exists a $k \in X(i, j) \cap Y(i, j)$, so $(i, j) \in R_1$.

Since both (i, k) and (k, j) are represented by hypothesis, there exist dummy paths $\pi(i, k)$ and $\pi(k, j)$. Therefore, the dummy path $\pi(i, j) = \pi(i, k)\ \pi(k, j)$ represents (i, j).

Otherwise, we have $(i, j) \in R_3$ and $Q_{pq} \in C'$ for some arc $(p, q) \in F(i, j)$. If $(p, q) = (i, k) \in F_1(i, j)$, then (i, j) contains (k, j) and since (k, j) is represented, there exists a dummy path $\pi(k, j)$. This path together with the arc (i, k) represents (i, j). A similar argument can be made to show (i, j) is represented if $(p, q) = (l, j) \in F_2(i, j)$. If $(p, q) = (l, k) \in F_3(i, j)$ then (i, j) contains both (i, l) and (k, j) and there exists dummy paths $\pi(i, l)$ and $\pi(k, j)$. These paths together with arc (l, k) represent (i, j). Thus D' is a solution to the dummy-arc problem.

Conversely, suppose D' is a solution to the dummy-arc problem with $|D'| \leq K + |R_2|$. By lemma 1 we can assume wlog that $D' \cap R_3 = \emptyset$. Let

$$C' = \{Q_{pq} \mid (p, q) \in D' \cap F\}.$$

Note that $|C'| \leq K$ because F does not include the dummy arc (i, j) for $(i, j) \in R_2$. To show that C' is a solution to the set-cover problem we need only verify that for every pair $(i, j) \in R_3$, there exists an arc

$$(p, q) \in D' \cap F(i, j).$$

Let $(i, j) \in R_3$. Then there exists a dummy path $\pi(i, j)$ in D', and $\pi(i, j) \neq (i, j)$, because otherwise $D' \cap R_3 \neq \emptyset$. Moreover, each intermediate node of $\pi(i, j)$ belongs either to $X(i, j)$, or $Y(i, j)$, but not both. Let (i, k) and (l, j) be the first and last arcs of $\pi(i, j)$. If $k \in X(i, j)$ then $(i, k) \in F_1(i, j)$. If $l \in Y(i, j)$ then $(l, j) \in F_2(i, j)$. The only remaining case is $k \in Y(i, j)$ and $l \in X(i, j)$. In this case the subpath of $\pi(i, j)$ leading from k to l must contain an arc (p, q) with $p \in Y(i, j)$ and $q \in X(i, j)$, and hence $(p, q) \in F_3(i, j)$. Thus, we have shown that some arc of $\pi(i, j)$ is in $F(i, j)$, which completes the proof. ■

From Theorem 1 we have immediately

COROLLARY 1: *The set-cover problem has an optimal solution of size K if and only if the dummy-arc problem has an optimal solution of size* $K + |R_2|$.

An optimal solution to the dummy-arc problem may be computed using the following algorithm:

1. Find the distinct pairs $(P^*(j), S^*(j))$ of the activity-set pairs $(P^*(S_v), S^*(s_v))$ and $(P^*(t_v), S^*(t_v))$. The activity-set pair for a node j is $(P^*(j), S^*(j))$.

2. Construct the framework by assigning to each activity v an arc (i, j) for which $P^*(i) = P^*(s_v)$ and $S^*(j) = S^*(t_v)$.

3. Add (i, j) to R if $i \neq j$ and $\exists\ (u, v) \in G$ such that $i = t_u$ and $j = s_v$.

4. For each pair $(i, j) \in R$, calculate the sets $X(i, j)$ and $Y(i, j)$, and use them to assign this pair to one of the subsets R_1, R_2, R_3.

5. For each pair $(i, j) \in R_3$, calculate $F(i, j)$ and add the pair (i, j) to all Q_{pq} sets for which $(p, q) \in F(i, j)$.

6. Find an optimal solution C' to the set-cover problem, *i.e.*, find the minimum number of Q_{pq} sets that cover R_3. The set of dummy arcs $\{(p, q) \mid Q_{pq} \in C'\} \cup R_2$ is an optimal solution to the dummy-arc problem.

For Figure 2 we have $R_3 = \{(2,6), (3,7)\}$, $Q_{47} = \{(3,7)\}$, $Q_{46} = \{(2,6)\}$, $Q_{45} = \{(2,6), (3,7)\}$, $Q_{35} = \{(3,7)\}$, $Q_{25} = \{(2,6)\}$, so an optimal solution is to choose Q_{45}. Therefore, the dummy arc (4,5), together with the dummy arcs (2,4), (3,4), (5,6), (5,7) and (6,8) defined by R_2, form a unique optimal solution to the dummy-arc problem.

The time complexity of the transformation (steps 1-5 of the algorithm) is $O(n^4)$, where n is the number of activities. Steps 1-4 can be done in time $O(n^3)$ (in step 4, begin by computing the set of feasible arcs: for each pair (i, j), check whether $P^*(i) \subset P^*(j)$ in $O(n)$ time), and step 5 is easily done $O(n^4)$. Although careful implementation can reduce the time for step 5 in typical cases, the worst case size of the set-cover problem is $\Omega(n^4)$ and this puts a lower bound on the time for step 5. For example, consider a set of activities $W \cup X \cup Y \cup Z$, where

$$W = \{w_0, \ldots, w_{k-1}\}, \qquad X = \{x_0, \ldots, x_{k-1}\},$$
$$Y = \{y_0, \ldots, y_{k-1}\}, \qquad Z = \{z_0, \ldots, z_{k-1}\},$$

and assuming k is even,

$$S(w_i) = Y \cup Z - \{y_i\} \quad \text{for} \quad i = 0, \ldots, k-1,$$
$$S(x_i) = \{y_i, \ldots, y_{((i-1+k/2) \bmod k)}\} \cup Z - \{z_i\} \quad \text{for} \quad i = 0, \ldots, k-1.$$

Given a particular i, we have $S(x_i) \subset S(w_j)$ for exactly $k/2$ values of j. All successor sets of W and X (predecessor sets of Y and Z) are distinct, so let

$$a_i = t_{w_i}, \qquad b_i = t_{x_i}, \qquad c_i = s_{y_i}, \qquad d_i = s_{z_i}.$$

There are $k^2/2$ pairs (b_i, c_j) in R. Each such pair is in $F_3(a_h, d_k)$ for $k^2/4$ pairs (a_h, d_k): (a_h, b_i) is feasible for exactly $k/2$ values of h and (c_j, d_k) is feasible for exactly $k/2$ values of k. Thus the total cardinality of all the F_3 sets is $k^4/8$ (recall that the number of activities is $4k$).

It is interesting to note that our reduction to the set-cover problem acts as an inverse of the reduction used by Krishnamoorthy and Deo [7] to prove the NP-completeness of the dummy-arc problem. Their proof reduces from an arbitrary instance of the vertex-cover problem to a particular instance of the dummy-arc problem. When our reduction is applied to their dummy-arc problem, we recover the original vertex-cover problem in its equivalent set-cover form.

4. SPECIAL CASES

The special cases of the dummy-arc problem that can be solved in polynomial time are those precedence relations that produce set-cover problems solvable in polynomial time. These special cases can be derived from either special cases of the set-cover problem, or from precedence relations that produce easy set-cover problems. In fact, most of the special cases that we address produce empty set-cover problems.

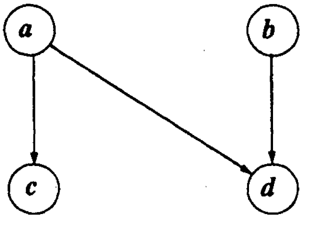

Figure 4

An AoN dag G is *N-free* (also *reversible* or a *line digraph*) if it does not contain the N-subgraph of Figure 4 as an induced subgraph. The case of N-free dags is well-known and has been characterized in several ways. The following characterization was used by Sysło [15] to show that N-free dags can be represented without *any* dummy arcs, which implies that $R = \emptyset$: A dag G is N-free iff for any two activities, u and v,

$$P(u) \cap P(v) \neq \emptyset \quad \text{implies} \quad P(u) = P(v);$$

or equivalently

$$S(u) \cap S(v) \neq \emptyset \quad \text{implies} \quad S(u) = S(v).$$

N-free dags can be generalized as follows. An AoN dag G is *frame-connected* if

(i) for every activity v with $S(v) \neq \varnothing$ there exists $w \in S(v)$ such that

$$P^*(w) = \bigcap_{z \in S(v)} P^*(z),$$

(ii) for every activity v with $P(v) \neq \varnothing$ there exists $u \in P(v)$ such that

$$S^*(u) = \bigcap_{z \in P(v)} S^*(z).$$

The term frame-connected is used to indicate that in the framework for G, every activity arc is on a path from $s_{\emptyset}$ to $t_{\emptyset}$. This can be seen when we note that a dag G is frame-connected iff (i) for every activity $v \in G$ with $t_v \neq t_{\emptyset}$, there exists some $w \in G$ such that $t_v = s_w$, and (ii) for every activity $v \in G$ with $s_v \neq s_{\emptyset}$, there exists some $u \in G$ such that $t_u = s_v$. (This follows from Theorem 1.2 of [1]). We now show that frame-connected dags define an empty set-cover problem, which implies the dummy-arc problem for frame-connected dags can be solved in polynomial time.

THEOREM 2: *If an AoN dag G is frame-connected, then R_3 is empty.*

Proof: Given $(i, j) \in R$, if $X(i, j) \cup Y(i, j) = \emptyset$, then $(i, j) \in R_2$. Otherwise, suppose $k \in X(i, j)$, so $k = t_x$ for some activity x because $(k, j) \in R$. Since G is frame-connected, there must also exist an activity y such that $k{=}s_y$, which implies $(i, j) \in R_1$. A symmetric argument can be made of $l \in Y(i, j)$. Thus $R_3 = \emptyset$. ■

The converse of theorem 2 is not true, however, as the example dag in Figure 5(a) illustrates. Figure 5(b), which gives an optimal AoA representation of Figure 5(a), clearly shows that this dag is not frame-connected, but defines an empty set-cover problem. Specifically,

$$R = \{(2,3), (2,4), (2,5), (3,5), (4,5)\} = R_1 \cup R_2,$$

where $R_1 = \{(2,5)\}$ and $R_2 = \{(2,3), (2,4), (3,5), (4,5)\}$.

A proper subclass of frame-connected dags is *interval orders*, where a precedence relation $\prec$ is an interval order if every activity v corresponds to an interval $(v^-, v^+] \subseteq \mathbf{R}$, and $u \prec v$ iff $u^+ < v^-$. The dag G for an interval order is derived from $\prec$ by transitive reduction. To show that an interval order is frame-connected, first note that $P^*(v) = \{u \mid u^+ < v^-\}$ and $S^*(\nu) = \{w \mid v^+ < w^-\}$. Also $P^*(u) \subseteq P^*(v)$ for all v with $u^- \leq v^-$, and $S^*(w) \subseteq S^*(v)$ for all w with $v^+ \leq w^+$. Since all activities can be ordered according to their l.h. endpoints, there must be some $w \in S(v)$ for

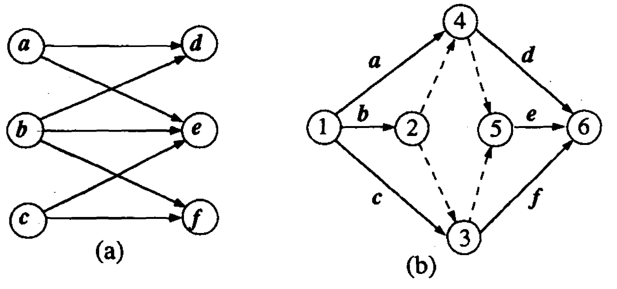

Figure 5

which w^- is farthest from v^+. Therefore, $P^*(w) \subseteq P^*(z), \ \forall z \in S(v)$ and $P^*(w) = \bigcap_{z \in S(v)} P^*(z)$. A symmetric argument can be made for the existence of a $u \in P(v)$ as required. Given the minimum number of nodes, the heuristic of Spinrad [13] finds an optimal solution in the special case of interval orders. Interval orders are incomparable with N-free dags. The forbidden subgraph $G_f = \{(a,c), (b,d)\}$ for interval orders is cleary N-free, while the forbidden subgraph for N-free dags (*Fig.* 4) is an interval order.

Marchioro *et al.* [8] introduced two classes of directed but not necessarily acyclic graphs called *adamant* and *inflexible*. A directed graph G is *adamant* if for any two activities, u and v,

$$S(u) \cap S(v) \neq \varnothing \quad \text{implies} \quad S(u) \subseteq S(v) \quad \text{or} \quad S(v) \subseteq S(u).$$

An adamant dag G is also *inflexible* if the reverse of G is also adamant, *i.e.*, if for any two activities u and v we also have

$$P(u) \cap P(v) \neq \varnothing \quad \text{implies} \quad P(u) \subseteq P(v) \quad \text{or} \quad P(v) \subseteq P(u).$$

It is easy to show that N-free dags are inflexible, and inflexible dags are frame-connected, where these inclusions are proper. However, the set of interval orders is incomparable with the set of inflexible dags. Other new classes of dags may be obtained by variations of the adamant conditions. We define *closure adamant* and *closure inflexible* dags by substituting $tc(G)$ for G in the previous definitions, so $S(\bullet)$ is replaced by $S^*(\bullet)$ and $P(\bullet)$ is replaced by $P^*(\bullet)$. Closure inflexible dags form a proper subset of frame-connected dags, while forming a proper generalization of both inflexible dags and interval orders.

We say a dag G is *anti-adamant* if the complementary condition holds, *i.e.*, if for any two activities u and v,

$$S(u) \cap S(v) \neq \varnothing \quad \text{implies} \quad S(u) \not\subseteq S(v) \quad \text{and} \quad S(v) \not\subseteq S(u).$$

Anti-inflexible, closure anti-adamant, and *closure anti-inflexible* dags are all similarly defined. Anti-inflexible dags generalize closure anti-inflexible dags (which must be bipartite) but are incomparable to frame-connected dags. Yet they also define empty set-cover problems. An interesting open question is an alternate characterization of all dags for which the set-cover problem is empty. Examples and further discussion can be found in Michael [9].

Another interesting open question is a characterization of precedence relations for which a greedy solution to the set-cover problem is optimal. A greedy solution is one in which the Q_{pq} subsets are first sorted in decreasing order of $|Q_{pq}|$, the size of the subsets, and then each subset is added to the (partial) solution provided it increases the number of elements covered. The heuristic of Spinrad [13] is similar to this greedy approach, except that the subsets are first sorted by the type of arc, and then by size. Spinrad showed that this heuristic finds and optimal solution for two-dimensional (2-D) partial orders, where a precedence relation is 2-D if every activity v corresponds to a point $(v_x, v_y) \in \mathbf{R}^2$, and $u \prec v$ iff $u_x < v_x$ and $u_y < v_y$. However, Michael [9] shows that it is not necessary to first sort by the type of arc. Hence, this class of dags can be included among those that have optimal greedy solutions.

Adamant dags, and their generalization as closure adamant dags, also belong to the class of dags for which greedy solutions are optimal. It is easy to show that for every activity v, either there exists an activity u such that $t_u = s_v$, or $s_v = s_\varnothing$. This implies $Y(i, j) = \varnothing$ for all $(i, j) \in R_3$. Furthermore, it is easy to show that for all $k \in X(i, j)$ there exists some $k' \in X(i, j)$ for which $S^*(k) \subset S^*(k')$. Hence there is an optimal solution that includes the arc (i, k'), the greedy choice. Adamant and closure adamant dags are incomparable with 2-D partial orders.

5. SUMMARY

We have presented an algorithm that reduces the NP-hard, dummy-arc problem to the set-cover problem in polynomial time. Thus optimal or heuristic solutions to the dummy-arc problem can be found using known methods of solving the set-cover problem. The reduction is derived from and is closely related to the Cantor-Dimsdale algorithm for constructing the minimum set of nodes: it is based on the predecessor and successor sets for the minimum set of nodes. Our analysis not only allowed us to show that many known special cases correspond to easy set-cover problems, but also to give efficient algorithms for entirely new families of special cases derived from adamant and inflexible dags.

ACKNOWLEDGEMENT

We thank the anonymous referees for their comments, which helped us to improve the presentation.

REFERENCES

1. D. G. Cantor and B. Dimsdale, On Direction-Preserving Maps of Graphs, *J. Comb. Theory*, 1969, *6*, pp. 165-176.
2. N. Christofides and S. Korman, A Computational Survey of Methods for the Set-Covering Problem, *Management Science*, 1975, *21*, pp. 591-599.
3. D. G. Corneil, C. C. Gotlieb and Y. M. Lee, Minimal Event-Node Network of Project Precedence Relation, *Comm. A.C.M.*, 1973, *16*, pp. 296-298.
4. B. Dimsdale, Computer Construction of Minimal Project Networks, *I.B.M. Syst. J.*, 1963, *2*, pp. 24-36.
5. A. C. Fisher, J. S. Liebman and G. L. Nemhauser, Computer Construction of Project Networks, *Comm. A.C.M.*, 1968, *11*, pp. 493-497.
6. M. Hayes, The Role of Activity Precedence Relations in Node-Oriented Networks, in *Project Planning by Network Analysis*, North-Holland, Amsterdam, 1969, pp. 128-146.
7. M. S. Krishnamoorthy and N. Deo, Complexity of the Minimum-Dummy-Activities Problem in a PERT Network, *Networks*, 1979, *9*, pp. 189-194.
8. P. Marchioro, A. Morgana, R. Petreschi and B. Simeone, Adamant Digraphs, *Discrete Math.*, 1988, *69*, pp. 253-261.
9. D. J. Michael, Optimal Representation of Activity Networks as Directed Acyclic Graphs, *Ph. D. Thesis,* 1991, North Carolina State University, Raleigh, NC 27695-7913.
10. M. Mrozek, A Note on Minimum-Dummy-Activities PERT Networks, *R.A.I.R.O. Rech. Oper.*, 1984, *18*, pp. 415-421.
11. M. Mrozek, Transitively Reduced and Transitively Closed Event Networks, *Networks*, 1989, *19*, pp. 47-72.
12. F. Sterboul and D. Wertheimer, Comment Construire un Graphe PERT Minimal, *R.A.I.R.O. Rech. Oper.*, 1980, *14*, pp. 85-98.
13. J. Spinrad, The Minimum Dummy Task Problem, *Networks*, 1986, *16*, pp. 331-348.
14. M. M. Sysło, Optimal Constructions of Event-Node Networks, *R.A.I.R.O., Rech. Oper.*, 1981, *15*, pp. 241-260.
15. M. M. Sysło, Optimal Construction of Reversible Digraphs, *Discrete Applied Mathematics*, 1984, *7*, pp. 209-220.

16. M. M. Sysło, A Graph-Theoretic Approach to the Jump-Number Problem, in *Graphs and Order*, I. Rival éd., D. Reidel, Dordrecht, 1985, pp. 185-215.
17. F. Vasko and G. Wilson, Hybrid Heuristics for the Minimum Cardinality Set Covering Problem, *Naval Res. Log. Quart.*, 1986, *33*, pp. 241-249.