Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule: An Application of Periodic Posets

JAMES B. ORLIN
Massachusetts Institute of Technology, Cambridge, Massachusetts
(Received June 1981; accepted November 1981)

In this paper we consider countably infinite partially ordered sets (posets) in which the order relations occur periodically. We show that the problem of determining the minimum number of chains (completely ordered subsets) needed to cover all of the elements may be solved efficiently as a finite network flow problem. A special case of the chain-cover problem for periodic posets is the problem of minimizing the number of individuals to meet a fixed periodically repeating set of tasks with set-up times between tasks. For example, if we interpret tasks as flights and individuals as airplanes, the resulting problem is to minimize the number of airplanes needed to fly a fixed daily repeating schedule of flights, where deadheading between airports is allowed.

IN THIS PAPER we consider and solve in polynomial time the problem of minimizing the number of vehicles to meet a fixed periodic schedule. For example, consider an airline that wishes to assign airplanes to a set of fixed daily repeating flights (e.g., San Francisco at 10:00 p.m. to Boston at 6:00 a.m.) in order to minimize the number of airplanes where deadheading between airports is allowed.

The finite horizon version of the above vehicle scheduling problem was solved by Dantzig and Fulkerson [1954]. The periodic version in which deadheading is forbidden was solved by Bartlett [1957] and by Bartlett and Charnes [1957]. (If deadheading is forbidden, the only question is how to start the schedule. Once in operation, a FIFO scheduling procedure is optimal.)

Dantzig [1962], in collaborative consulting work for United Airlines, solved the problem of minimizing the number of vehicles to meet a fixed periodic schedule under the restriction that the schedule of deadheading flights also repeats daily. Simpson [1969] describes Dantzig's solution technique (along with a host of solution techniques for related problems), and Orlin [1981a] proves that Dantzig's solution is also optimal if the stationarity constraint on deadheading is relaxed. Recently Wollmer

Subject classification: 432 an application of periodic posets, 484 periodic vehicle scheduling via network flows, 832 the minimum number of vehicles to meet a periodic schedule.

Operations Research
Vol. 30, No. 4, July–August 1982
0030-364X/82/3004-0760 $01.25

[1980] independently discovered an alternative approach for solving the airplane scheduling problem. His approach is similar to the one presented here in that it is based on finding chain-covers for partially ordered sets. Wollmer's technique is to find an optimal schedule for two consecutive days and to extend that schedule to an infinite-horizon schedule.

Finally, Orlin [1981a] has solved the more general problem of minimizing the average linear cost per day of flying a schedule subject to a fixed number of airplanes. This last paper involves a solution technique substantially different from the technique presented here, although both involve solutions induced from finite minimum-cost network flows.

The periodic vehicle scheduling problem may be expressed in terms of task scheduling as follows: What is the minimum number of individuals to meet a fixed periodically repeating set of tasks? (For airplane scheduling, the tasks are flights and the individuals are airplanes.) In Section 4 this problem is shown to be a special case of the "minimum chain-cover problem for periodic partially ordered sets." This latter problem is formulated in Section 1. In Section 2, we solve the above mentioned chain-cover problem in the special case that the periodic poset is "forward or backward dominating." In this case, an optimal chain-cover is induced by an optimal cycle-cover of a related finite graph. In Section 3, we solve the general chain-cover problem for periodic posets in two phases. The first phase consists of solving two distinct cycle-cover problems, and the second phase consists of solving a bipartite matching problem.

These results generalize the work of Ford and Fulkerson [1956] who showed that the finite version of the above task scheduling problem may be solved as a special case of the minimum chain-cover problem for finite partially ordered sets.

Periodic Partially Ordered Sets

A partially ordered set (poset) is a set with a transitive antisymmetric order relation $>$. A "periodic poset" is a special type of infinite poset in which the "relations" occur periodically. A "chain" is a set of elements every two of which are related, and a "chain-cover" is a partition of the elements of the partially ordered set into chains. Periodic posets are highly structured, and in Section 2 we exploit this structure to obtain a polynomial time algorithm that determines a minimum cardinality chain-cover for a subclass of periodic posets. The technique is similar to that used by Orlin [1981a] to obtain maximum-throughput dynamic network flows, in that the solution is obtained by reinterpreting a minimum-cost flow in a related finite network.

Periodic Interval Graphs and Circular Arc Graphs

An *intersection graph* is a graph whose vertices are associated with subsets of a set, and two vertices are adjacent if and only if the corresponding sets have a nonempty intersection. A *periodic interval graph* is an intersection graph in which the associated subsets are intervals that are periodically spaced over the real line. A *circular arc graph* is an intersection graph in which the associated subsets are arcs on a circle. The problems of coloring circular arc graphs and (finite) interval graphs have both been studied extensively, e.g., Garey et al. [1980], Golumbic [1980], Lekkerkerker and Boland [1962], Orlin et al. [1981], and Tucker [1975].

In Section 4, we show that the coloring problem for periodic interval graphs is a special case of the task scheduling problem, and thus it is solvable as a network flow problem. We also observe that the circular arc coloring problem, which was proved NP-complete by Garey et al. is a special case of the task scheduling problem under the added restriction that each instance of the same task is carried out by the same person, proving that this latter problem is NP-hard. These results contrast with a recent result by Orlin [1981b] showing that the coloring problem for "periodic graphs" is polynomial-space complete.

1. PERIODIC POSETS AND DILWORTH'S THEOREM

A *partially ordered set* (*poset*) is a set with a transitive, antisymmetric relation $\succ$. A *chain* of a poset P is a (possibly infinite) subset of elements of P that are pairwise related, and an *anti-chain* is a subset of elements of P that are pairwise unrelated. A *chain-cover* of P is a partition (or decomposition) of the elements of P into chains. It is obvious that the number of chains in any chain-cover is at least the number of elements in any anti-chain. This inequality leads to the min-max result proved by Dilworth [1950].

DILWORTH'S THEOREM. *Let P be a finite or countably infinite partially ordered set; then the minimum cardinality of a chain-cover is the maximum cardinality of an anti-chain.*

Let N be a finite index set, let Z be the set of integers, and let $P = \{i^r : i \in N, r \in Z\}$ be a partially ordered set. We say that P is *periodically closed* if it satisfies relation (1) below.

$$i^p \succ j^r \quad \text{if and only if} \quad i^{p+1} \succ j^{r+1}. \tag{1}$$

If $P = \{i^r : i \in N, r \in Z\}$ and S is any relation on P (not necessarily a partial order), then the relation induced by S and (1) is called the *periodic closure* of S. For example, the periodic closure of the singleton set $\{i^r \succ j^p\}$ is the set of related pairs $\{i^k \succ j^{k+p-r} : k \in Z\}$. The *transitive closure* of S is the set of related pairs $u \succ v$ such that there is a finite sequence $u = u_1, \cdots, u_k = v$ of elements of P such that "$u_i \succ u_{i+1}$" is a related pair of S for $i = 1, \cdots, k - 1$. The *transitive-periodic closure* of S is the transitive closure of the periodic closure of S.

REMARK 1. *The transitive-periodic closure is periodically closed.*

We say that P is a *periodic partially ordered set* if P is a partially ordered set with elements $\{i^r : i \in N, r \in Z\}$, and the set of partial order relations is the transitive-periodic closure of some finite set.

Example 1 (A periodic poset). Let $P = \{1^r, 2^r : r \in Z\}$ such that (1) $1^r > 1^p$ for $r > p$, and (2) $2^r > 2^p$ for $r < p$, and (3) $1^r > 2^p$ for $r, p \in Z$. Then this partially ordered set is the transitive-periodic closure of the set $\{1^1 > 1^0, 2^0 > 2^1, 1^0 > 2^0\}$.

Example 2 (A partially ordered set that is periodically closed but is not a periodic poset). Let $P = \{1^r, 2^r : r \in Z\}$ such that $1^r > 2^p$ for all $r, p \in Z$, and all other elements are unrelated. It is clear that P is not the transitive-periodic closure of a finite set of relations.

If the periodic poset P is the transitive-periodic closure of relation S, we say that S *generates* P and that S is a *generating relation* of P.

The main theoretical result of this paper is a polynomial time algorithm for finding a minimum cardinality chain-cover and a maximum cardinality anti-chain in a periodic poset. Here, polynomial time means that the number of elementary operations to determine the chain-cover and anti-chain is polynomially bounded in the length of the input for the generating set (which we assume is given as input) and in the cardinality of the chain-cover and anti-chain (which are assumed to be finite).

An Alternative Description of Periodic Posets

Let P be a partially ordered set whose elements are the set of integers. We say that P is *periodic with period n* if (1′) holds.

$$i > j \quad \text{if and only if} \quad i + n > j + n. \tag{1′}$$

There is a 1:1 relation between the sets $P' = Z$ satisfying (1′) and partially ordered sets $P = \{i^r : i = 1, \cdots, n, r \in Z\}$ satisfying (1), which is as follows: we associate the element $i^r \in P$ with element $i + rn \in P'$.

Of course, if P is periodic with period n, then it is also periodic with period kn for any positive integer k. Moreover, it is easy to show that if P is periodic with periods n and n' then it is also periodic with period k, where k is the greatest common divisor of n and n'.

If we consider partially ordered sets with $P' = Z$ as above, the notation is somewhat simpler. However, we will continue using the previous notation because it helps to make the proofs and applications more transparent.

2. FINDING MINIMUM CARDINALITY CHAIN-COVERS AND MAXIMUM CARDINALITY ANTI-CHAINS IN FORWARD-DOMINATING PERIODIC POSETS

Forward and Backward Dominating Indices

In a periodic poset, index i is said to be *forward dominating* (resp., *backward dominating*) if there is an integer $p > 0$ (resp., $p < 0$) such that $i^0 > i^p$.

REMARK 2. *If some index i of a periodic poset P is neither forward nor backward dominating, then the set $\{i^r : r \in Z\}$ is an anti-chain of P.*

REMARK 3. *No index of a periodic poset is both forward and backward dominating.*

Proof. Let p and r be positive integers. If $i^0 > i^{-p}$ then $i^{rp} > i^0$. If $i^0 > i^r$ then $i^0 > i^{rp}$. It is therefore impossible that both $i^0 > i^{-p}$ and $i^0 > i^r$.

If index i of a periodic poset is forward (resp., backward) dominating then we also say that element i^p is *forward* (resp., *backward*) *dominating* for all $p \in Z$. If every element of the periodic poset P is forward (resp., backward) dominating then we say that P is *forward* (resp., *backward*) *dominating.*

An Overview

Below we give a polynomial time algorithm for finding a minimum cardinality chain-cover in either a forward or backward dominating periodic poset P. To do this, we associate a finite network with each generating relation for P, and we show that certain periodic chains may be induced from the cycles in the finite network. Finally we show that a minimum cardinality chain-cover of P may be induced from an optimal covering of the nodes of the associated finite network by directed cycles. At the same time we determine a maximum cardinality anti-chain in P. These results are applied in Section 4 to solve the periodic task scheduling problem.

Suppose $P = P^* \cup P'$, where P^* (resp., P') is the subset of forward (resp., backward) dominating elements of periodic poset P. The union of the minimum chain-covers for P^* and P' does not, in general, form a minimum chain-cover for P. In Section 3, we show how to transform this union of chain-covers into a minimum cardinality chain-cover by pairing a number of chains.

Generating Networks

Let P be a periodic poset with element set $\{i^r : i \in N, r \in Z\}$, and let S be a finite generating relation for P. We associate with S a generating

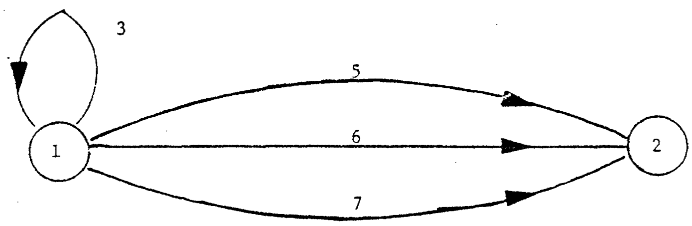

Figure 1. A generating network.

network $G^S = (N, A^S)$, where N is the set of nodes and A^S is an arc set that is constructed as follows: for each ordered pair $i^r > j^p$ in S, there is an associated arc a in A^S directed from i to j and with an associated *length* $c_a = p - r$. Of course, G^S may contain multiple arcs.

Example 3. Let $P = \{1^r, 2^r : r \in Z\}$ be a periodic poset such that (1) $1^r > 2^p$ if $p - r \geq 5$, (2) $1^r > 1^p$ if $p - r = 3k$ for some positive integer k and (3) all other elements are unrelated. Then P is generated by the set of related pairs $S = \{1^0 > 1^3, 1^0 > 2^5, 1^0 > 2^6, 1^0 > 2^7\}$. The generating network G^S is portrayed in Figure 1. The numbers on the arcs are the arc lengths.

A *directed path* in network G is an alternating sequence $i_1, a_1, \cdots, a_{k-1}, i_k$ of nodes and arcs such that arc a_j is directed from node i_j to node i_{j+1} for $j = 1, \cdots, k-1$. The *length* of a directed path is the sum of the lengths of the arcs of the path. A *directed cycle* is a directed path in which the initial node is the same as the terminal node, *and* no other node appears twice on the path. Figure 2 represents a directed path of length 6. Figure 3 is a directed cycle of length 3.

LEMMA 1. *Let P be a periodic poset with generating relation S. Then $i^r > j^p \in P$ if and only if there is a path in the generating network G^S from node i to node j with length $p - r$.*

Proof. Let $PC(S)$ be the periodic closure of set S. It is clear that $i^r > j^p \in PC(S)$ if and only if there is an arc $a = (i, j) \in A^S$ with length $c_a = p - r$. By definition of the transitive-periodic closure, $i^r > j^p$ in P if and only if there is a finite sequence of elements $i^r = i_1^{r_1}, \cdots, i_k^{r_k} = j^p$ in P such that the related pair $i_l^{r_l} > i_{l+1}^{r_{l+1}}$ is in $PC(S)$ for $l = 1, \cdots, k-1$. The above is true if and only if there are corresponding arcs a_l from i_l to i_{l+1} with lengths $r_{l+1} - r_l$, and this is true if and only if the path $i_1, a_1, \cdots, a_{k-1}, i_k$ is a path in G^S from i to j with length $p - r$.

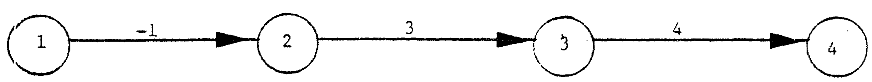

Figure 2. A directed path of length 6.

COROLLARY 1. *Let C be a simple cycle with length r in the generating network G^S. Let P' be the periodic poset generated by C. Then the elements of P' partition into $|r|$ chains such that no element in one chain is related to an element in another chain.*

Proof. Let the cycle be a path from i to i with length r. Then C induces a chain in P' from i^p to i^{p+r} for all integers p. Let S_j denote the chain obtained by concatenating the chains from i^p to i^{p+r} for $\{p:p = rk + j, k \in Z\}$. Then the chains $S_1, \cdots, S_{|r|}$ cover all elements l^p for node l of C and $p \in Z$. Since all paths in C from i to i have a length that is an integral multiple of r, it follows from Lemma 1 that elements i^j and i^k in distinct chains are unrelated. Therefore, any two elements in distinct chains are unrelated.

Example 4. Let P be the periodic poset whose generating network is portrayed in Figure 3. Then the elements of P decompose into the three chains $S_j = \{1^{j+3k}, 2^{j+1+3k}, 3^{j+2+3k}, 4^{j+2+3k}, 5^{j+1+3k}: k \in Z\}$ for $j = 1, 2, 3$.

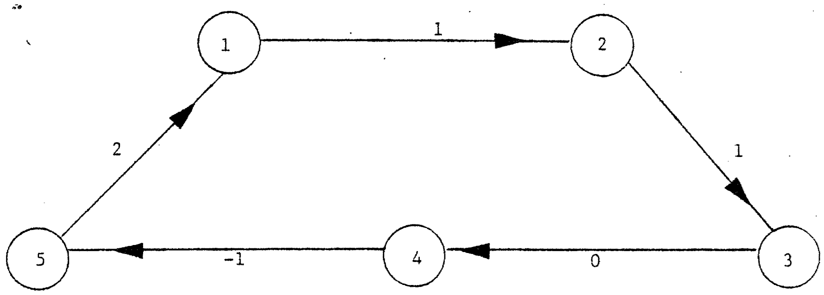

Figure 3. A directed cycle of length 3.

Let $G = (N, A)$ be a generating network for a periodic poset. A *cycle-cover* for G is a union of cycles (not necessarily disjoint) that contains all of the nodes of G. The *length* of a cycle-cover is the sum of the lengths of the cycles. The following corollary is an immediate consequence of Corollary 1, and the fact that all directed cycles of a generating network for a forward-dominating periodic poset have positive length.

COROLLARY 2. *Let G be the generating network for a forward-dominating periodic poset P, and let $\mathscr{C}$ be a cycle-cover of G. Then $\mathscr{C}$ induces a chain-cover of P whose cardinality is the length of the cycle-cover $\mathscr{C}$.*

To obtain a minimum length cycle-cover of network G, it suffices to solve the network flow problem (2) described below.

$$\text{Minimize} \qquad z = \textstyle\sum_{a\in A} c_a x_a \tag{2.1}$$

$$\text{Subject to} \quad \textstyle\sum_{a\in T_i} x_a - s_i = 0 \quad \text{for} \quad i \in N, \tag{2.2}$$

$$s_i - \textstyle\sum_{a\in H_i} x_a = 0 \quad \text{for} \quad i \in N, \tag{2.3}$$

$$s_i \geq 1 \quad \text{for} \quad i \in N, \tag{2.4}$$

$$x_a \geq 0 \quad \text{for} \quad a \in A, \tag{2.5}$$

where T_i (resp., H_i) is the set of arcs of G whose tail (resp., head) is node i.

We see that an optimum solution to (2) induces a minimum length chain cover as follows. First, we note that the linear program (2) is feasible since each node is in some directed cycle because P is forward dominating, and all cycles have positive length also because the poset is forward dominating. Second, each feasible solution to (2) is a nonnegative circulation with the property that the flow into node i is $s_i \geq 1$. Moreover, each basic solution is integer-valued since (2) is a network flow problem. It is well known (see for example Ford and Fulkerson [1962]) that a nonnegative integral circulation may be decomposed into the sum of unit flows around cycles, and this decomposition gives the desired cycle-cover.

Just as chain-covers are closely related to cycle-covers, so are anti-chains closely related to the dual of cycle-covers, defined below.

A *node-assignment* of a directed graph G is an assignment of non-negative integer values to each node. If $y = (y_i)$ is a node-assignment then the *cost* of y is $y_1 + \cdots + y_n$. A node-assignment is called *cycle-dominated* if for each directed cycle C we have

$$\textstyle\sum_{i \in N(C)} y_i \leq \sum_{a \in A(C)} c_a \tag{3}$$

where $N(C)$ and $A(C)$ are the nodes and arcs of cycle C.

COROLLARY 3. *Let P be a forward-dominating periodic poset and let G be a generating network for P. Then each anti-chain A of P induces a cycle-dominated node-assignment $y = (y_i)$ defined as follows:*

$y_i =$ the number of elements of A whose index is i.

Proof. Suppose that C is a directed cycle of G and let r be the length of C. Relabel the nodes of G (and thus the elements of P) so that the nodes of the cycle are $1, \cdots, k$. Let P' be the subset of P consisting of elements $\{i^p : i = 1, \cdots, k \text{ and } p \in Z\}$. Then by Corollary 1, the elements of P' partition into r chains and thus P' has no anti-chain with cardinality greater than r. Therefore

$$\textstyle\sum_{i \in N(C)} y_i = \sum_{i=1}^{k} y_i \leq \sum_{a \in A(C)} c_a.$$

THEOREM 1. *Let P be a forward-dominating periodic poset and let G be a generating network for P. Then the following four quantities are equal:*

$q_1 =$ the minimum cardinality of a chain-cover of P
$q_2 =$ the maximum cardinality of an anti-chain of P
$q_3 =$ the minimum length of a cycle-cover of G
$q_4 =$ the maximum cost of a cycle-dominated node-assignment of G.

Proof. We first show the following weak duality inequalities: $q_2 \le q_1$, $q_4 \le q_3$. The inequality $q_2 \le q_1$ is the obvious weak duality version of Dilworth's Theorem. The inequality $q_1 \le q_3$ is a consequence of Corollary 2, and $q_2 \le q_4$ is a consequence of Corollary 3. Finally, we see that $q_4 \le q_3$ as follows. Suppose that $y = (y_i)$ is a cycle-dominated node-assignment, and that $\mathscr{C}$ is a cycle-cover with length k. Then

$$\textstyle\sum_{i=1}^{n} y_i \le \sum_{C \in \mathscr{C}} \sum_{i \in N(C)} y_i \le \sum_{C \in \mathscr{C}} \sum_{a \in A(C)} c_a = k.$$

To show that all four quantities q_1, q_2, q_3, q_4 are equal, it suffices now to show that $q_2 \ge q_3$. To this end, we consider the linear program (4) described below, which is the dual to linear program (2).

$$\text{Maximize} \quad w = \textstyle\sum_{i \in N} y_i \tag{4.1}$$

$$\text{Subject to} \quad -u_i + v_j \le c_a \quad \text{for} \quad a = (i, j) \in A, \tag{4.2}$$

$$-u_i + v_i - y_i \ge 0 \quad \text{for} \quad i \in N, \tag{4.3}$$

$$y_i \ge 0 \quad \text{for} \quad i \in N. \tag{4.4}$$

Because (2) has an optimal solution x^*, linear program (4) also has an optimal solution (u, v, y) and by linear programming duality the objective value of (u, v, y) is q_3. Furthermore, since (4) is the dual of a network flow problem, we may assume that (u, v, y) is an integer-valued basic solution. Let

$$S = \cup_i \{i^{u_i}, \cdots, i^{v_i - 1}\}.$$

Then $|S| = y_1 + \cdots + y_n = q_3$. We now claim that S is an anti-chain, and this proves the theorem.

Suppose there is a chain containing two elements i^r, j^p of S. By Lemma 1, there is a path in G from node i to node j of length $p - r$, and suppose it is the path $i = i_1, a_1, \cdots, a_{k-1}, i_k = j$. Then the length of the path may also be written as follows:

$$\begin{aligned} p - r &= \textstyle\sum_{l=1}^{k-1} c_{a_l} \ge \sum_{l=1}^{k-1} (v_{i_{l+1}} - u_{i_l}) \\ &= v_j - u_i + \textstyle\sum_{l=2}^{k-1} (v_{i_l} - u_{i_l}) \ge v_j - u_i. \end{aligned}$$

If $i^r \in S$ then $r \ge u_i$ and hence $p \ge v_j$, contradicting that $j^p \in S$, and proving that S is an anti-chain.

The result above that q_3 is equal to q_4 is a special case of a combinatorial result of Cameron and Edmonds [1980]. We have included that part of the proof of Theorem 1 for completeness. In fact, Cameron and Edmonds show that the maximum *linear* cost of a cycle-dominated node-assignment is equal to the minimum length of a "generalized" cycle-cover. (The terminology used here is different from that of Cameron and Edmonds.)

In order to find maximum cardinality anti-chains and minimum cardinality chain-covers in backward dominating periodic posets, it is easy to transform the poset into a forward dominating poset as follows.

REMARK 4. *Let P be a backward-dominating periodic poset with generating relation S. Let P' be the forward-dominating periodic poset derived from P by replacing each related pair $i > j$ with the pair $j > i$. Then P' is generated by the relation S' obtained from S by reversing the direction of all related pairs. Furthermore, each chain* (*resp., anti-chain*) *of P is a chain* (*resp., anti-chain*) *of P'.*

At this point the following conjecture may seem plausible: "A minimum cardinality chain-cover for a periodic poset P may be obtained by finding minimum cardinality chain-covers for the forward-dominating elements and the backward-dominating elements, and then taking the union." This decomposition approach fails because, as in Example 1, sometimes a chain of forward-dominating elements may be paired with a chain of backward-dominating elements to yield a single chain. A decomposition argument plus a pairing approach is given in the next section to determine a minimum chain-cover for periodic posets with both forward and backward dominating elements.

3. MINIMUM CARDINALITY CHAIN-COVERS IN PERIODIC POSETS

In Section 1 we showed how to obtain in polynomial time a minimum cardinality chain-cover and a maximum cardinality anti-chain in a forward (resp., backward) dominating periodic poset. The proof depended critically on the fact that all cycles of a generating network have positive (resp., negative) length. In this section, we show how to extend the results to periodic posets that are neither forward nor backward dominating.

The gist of the procedure is as follows. Partition the elements of P into sets P' and P^* of forward and backward dominating elements. Then find minimum cardinality chain-covers S' and S^* for P' and P^* using the procedure in Section 2. A forward-dominating chain C and a backward-dominating chain D are *compatible* if $C \cup D$ is a chain. A minimum cardinality chain-cover may be obtained by taking $S' \cup S^*$ and then pairing as many compatible pairs of chains as is possible. This procedure may be carried out by solving a matching problem on an associated bipartite graph that has $|S'| + |S^*|$ vertices, as detailed below.

Henceforth, we consider only those periodic posets in which each element is either forward or backward dominating. By Remark 2, the minimum cardinality of a chain-cover in all other periodic posets is ∞.

We say that a pair C, D of chains is *compatible* if $C \cup D$ is a chain. To *pair* two compatible chains C and D is to create the single chain $C \cup D$.

THEOREM 2. *Let $P = P' \cup P^*$ be a periodic poset with forward (resp., backward) dominating subset of elements P' (resp., P^*). Let S' and S^* be minimum cardinality chain-covers for P' and P^* induced by cycle-covers of the corresponding generating networks. Then a minimum cardinality chain-cover for P may be obtained from $S' \cup S^*$ by pairing a maximum number of compatible chains.*

Proof. As a preliminary we prove the following lemma.

LEMMA 2. *If C' and C^* are chains of S' and S^* respectively, then either C' and C^* are compatible or else no element of C' is related to an element of C^*.*

Proof. Suppose $i^r \in C'$, $j^p \in C^*$ and $i^r > j^p$. (The proof in the case that $j^p > i^r$ is symmetric to the one below.) Chains C' and C^* were induced by cycles in the corresponding generating networks as in Corollary 1. Let l and $-k$ be the lengths of these cycles. It follows that for each integer t we have $i^{r+lkt} \in C'$ and $j^{p+lkt} \in C^*$. Because P is periodic, we also have $i^{r+lkt} > j^{p+lkt}$ for each integer t. Then for every $u \in C'$ and $v \in C^*$, we can choose t sufficiently large so that $u > i^{r+lkt} > j^{p+lkt} > v$.

To complete the proof of Theorem 2, we let $U = \{u_1, \cdots, u_r\}$ (resp., $V = \{v_1, \cdots, v_p\}$) be a maximum cardinality anti-chain of P' (resp., P^*). Of course, $r = |S'|$, $p = |S^*|$. Furthermore, for every $s \in U \cup V$ there is a unique associated chain C in $S' \cup S^*$ such that $s \in C$ because $|S'| = |U|$ and $|S^*| = |V|$.

Let B be a bipartite graph with vertex set $U \cup V$, and where u_i is adjacent to v_j if the corresponding elements of P are related. Let I be a maximum cardinality independent set in B, and let M be a maximum cardinality matching. By our construction, I is an anti-chain of P. In fact, we claim that I is a maximum cardinality anti-chain. To see this, let S denote the chain-cover of P derived from $S' \cup S^*$ by pairing compatible chains C' and C^* whenever the associated vertices of B are matched. (This pairing is legal by Lemma 2.) The cardinality of this chain-cover is $|S'| + |S^*| - |M|$, which is equal to $|I|$ by a direct extension of the König-Egerváry duality theorem for bipartite matchings. Therefore, the chain-cover S has minimum cardinality, and the anti-chain I has maximum cardinality.

4. THE MINIMAL NUMBER OF INDIVIDUALS TO MEET A FIXED PERIODIC SCHEDULE OF TASKS

Ford and Fulkerson [1962] showed how to find the minimum number of individuals needed to meet a fixed schedule of tasks by reducing that problem to a special case of finding a minimum cardinality chain-cover in a poset. Here we extend their results to the case in which there are a finite number of tasks that must be performed periodically over an infinite horizon.

Let $T_1, \cdots, T_n$ be a set of tasks that must be carried out periodically, and let p denote the period length. Associated with task T_i are nonnegative real numbers a_i and b_i such that T_i must be processed by an individual during the time interval $(a_i + kp, b_i + kp)$ for $k = 0, 1, 2, \cdots$. We refer to the kth iteration of task T_i as the *kth instance* of T_i. (We allow that $a_i + p < b_i$, which corresponds to the case that two instances of the same task are processed in overlapping intervals.) The individuals (processors) are identical, and thus any individual can carry out any task. Finally, there is a set-up time r_{ij} between the successive processings of instances of task T_i and task T_j. We assume that $r = (r_{ij})$ satisfies the following inequalities: $r_{ik} \leq r_{ij} + r_{jk} + b_j - a_j$. Thus the set-up time between T_i and T_k is no greater than the set-up time between T_i and T_j, followed by the processing of T_j and the set-up between T_j and T_k.

We transform the task scheduling problem into the chain-covering problem as follows. Let j^r denote the rth instance of task T_j, which is carried out in interval $(a_j + pr, b_j + pr)$. We then induce a periodic partial order as follows: $i^q > j^s$ if the qth instance of task T_i may be a predecessor of the sth instance of task T_j; i.e., $b_i + pq + r_{ij} \leq a_j + ps$. So long as r_{ii} is finite for each i, it is clear that the above periodic poset is forward dominating. Furthermore, each chain of P is a collection of instances of tasks that can be carried out by the same individual. Thus the minimum number of individuals needed to carry out all of the tasks is the minimum cardinality of a chain-cover of the induced periodic poset. Furthermore, we have the following interpretation of Dilworth's theorem:

> The minimum number of individuals needed to carry out a fixed number of periodically repeating tasks is equal to the maximum number of instances of tasks such that no two of them may be carried out by the same individual.

If we add the restriction that each task is periodically carried out by the same individual, then the resulting problem is NP-hard, as demonstrated in a subsection below on "circular arc graphs."

An Application to Airplane Scheduling

Consider an airline that must schedule a minimum number of airplanes to meet a fixed daily-repeating set of flights, where deadheading is permitted. This problem is easily seen to be a special case of the above task scheduling problem. The tasks are flights, and the period length is one day. The set-up time r_{ij} is the required delay between the arrival of flight i and the departure of flight j, assuming that the same airplane flies both flights. We allow that the arrival site s for flight i is different from

TABLE I
DAILY REQUIRED FLIGHTS

Flight No.	Depart	Arrive
1	Honolulu 1:00 p.m.	Washington, D.C. 11:00 p.m.
2	New York 3:00 p.m.	Tokyo 4:00 a.m.
3	London 1:00 p.m.	Paris 2:00 p.m.

the departure site t for flight j, in which case the set-up time r_{ij} would include the deadhead time from airport s to airport t.

Various versions of the above problem have been considered in the literature. Dantzig and Fulkerson solved the problem of minimizing the number of vehicles to meet a fixed finite-horizon schedule. (This was Fulkerson's first paper on network flows.) The problem of minimizing the number of vehicles to meet a periodic schedule in which all routes are required—so deadheading is not allowed—has been solved by Bartlett, and Bartlett and Charnes, and the problem has been applied to railroad scheduling.

Dantzig considered various airline scheduling problems including the above problem of minimizing the number of airplanes to meet a fixed periodic schedule under the added restriction that the final flight schedule consists of daily-repeating flights; e.g., if a plane deadheads from Boston to New York on one day, then it must deadhead from Boston to New York every day, and at the same time. Dantzig's technique, as described by Simpson is similar to the one described below except that Dantzig's finite network flow problem is a minimum cost circulation problem in which the flights are represented by arcs instead of nodes. Orlin [1981a] shows that Dantzig's technique is also optimal if we relax the restriction that the final flight schedule repeats daily.

As Dantzig and his collaborators observed, in order to satisfy the stationary flight schedule each airplane flies a periodic schedule that may not repeat daily, but will repeat after a number of days. We illustrate below the technique for determining a minimum number of vehicles to meet a fixed periodic schedule.

Recently Wollmer has independently given an efficient algorithm for the airplane scheduling problem. His technique, based on Dilworth's theorem, is to show how to extend a schedule that is optimal for two days into one that is optimal for an infinite horizon schedule.

TABLE II
FLIGHT TIMES IN HOURS

	Honolulu	London	New York
Paris	15	1	7
Tokyo	8	12	13
Washington	10	7	1

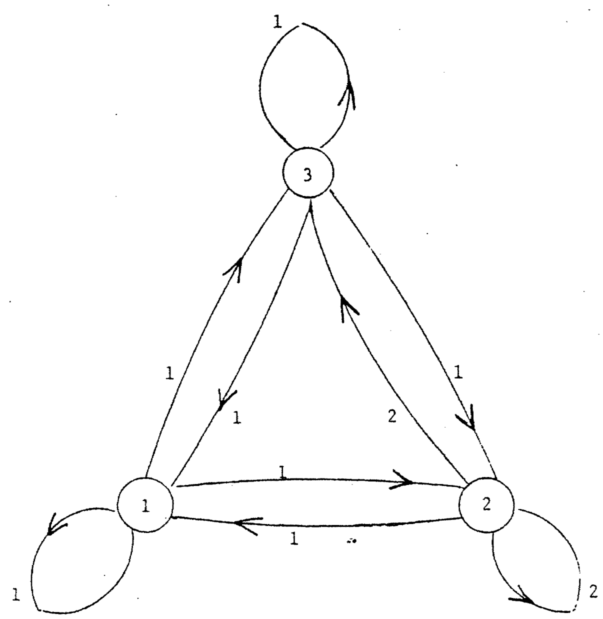

Figure 4. A generating network for the periodic poset of an airplane scheduling problem.

Example. Consider the airplane scheduling problem in which the three daily required flights are given in Table I. In order for a given plane to get from one flight to the next, it must deadhead from the arrival site of the one flight to the departure site of the next flight, and these travel times are given in Table II. The elements of the periodic poset are $\{1^p, 2^p, 3^p: p = 1, 2, 3, \cdots\}$ where i^p represents flight i on day p. A generating network is portrayed in Figure 4, and a minimum cycle-cover is portrayed

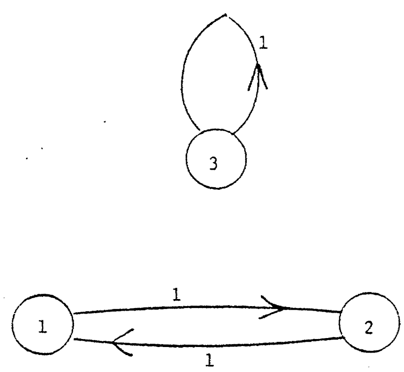

Figure 5. A minimum length cycle-cover for the generating network shown in Figure 1.

in Figure 5. (In the flight schedule, all departure times and arrival times are in Eastern Standard Time instead of local times.)

Thus the minimum number of airplanes to meet this schedule is three. The first plane travels round trip from London to Paris daily. The second and third planes make cyclic trips from Honolulu to Washington to New York to Tokyo to Honolulu, and each of these trips repeats every two days. (The solution is not unique because the directed cycle 3, 2, 1, 3 is also a minimum length cycle-cover.) Since on any given day the three flights must be flown by three different planes, it follows that the minimum number of planes to fly the schedule is three. In fact, Dilworth's theorem implies:

> The minimum number of airplanes to meet a fixed periodic schedule is the maximum number of instances of flights, no two of which may be flown by the same airplane.

Coloring Periodic Interval Graphs

Another "application" of the task scheduling problem is to a coloring problem in graph theory. An *intersection graph* G is the graph derived from a collection of subsets S as follows. We associate each vertex of G with a subset in S, and two vertices of G are adjacent if the corresponding subsets have a nonempty intersection. An *interval graph* is the intersection graph of a set of intervals on the real line. An interval graph is *periodic* if the corresponding set of intervals is an infinite set spaced periodically over the real line, i.e., the set of intervals may be written as follows: $\{(a_i + kp, b_i + kp) : i \in N, k \in Z\}$. In other words, it is the set of intervals in which tasks may be carried out for the periodic task scheduling problem.

Interval graphs were introduced into the literature by Lekkerkerker and Boland, and have been studied extensively. For a recent book that surveys the literature on interval graphs see Golumbic.

It is easy to see that the minimum number of colors needed to color the vertices of a periodic interval graph is exactly the number of individuals needed to carry out the tasks of the corresponding task scheduling problem, assuming $r_{ij} = 0$ for all i, j.

Coloring Circular Arc Graphs

Consider the problem of coloring periodic interval graphs with the added restriction that the set of vertices corresponding to the intervals $\{(a_i + kp, b_i + kp) : k \in Z\}$ be colored the same color for any fixed i. We may reinterpret the coloring as follows. Consider a circle whose points are real numbers in the interval $(0, p)$ extending clockwise around the circle. Let (a, b) denote an arc of the circle extending clockwise from point a to point b, and let $S = \{(a_i, b_i): i \in N\}$. Then the intersection graph for S is a *circular arc graph*, and any k-coloring of the graph may be extended to a k-coloring of the periodic interval graph such that $(a_i + kp, b_i + kp)$ is given the same color for each p.

In terms of the scheduling problem, such a coloring corresponds to an assignment of tasks to individuals so that each instance of task i is assigned to the same individual, or each instance of a flight is flown by the same plane. Recently Garey et al. proved that the problem of coloring circular graphs is NP-hard. Since the circular arc coloring problem is a special case of the airplane scheduling problem with the restriction that schedules for each airplane repeat daily, this latter problem is also NP-hard.

There have been partial results on coloring circular arc graphs. For example, Tucker analyzes several heuristics and reduces the coloring problem to a multicommodity flow problem. Recently Orlin et al. gave a polynomial time algorithm for the special case of circular arc coloring in which no arc is contained within another.

ACKNOWLEDGMENT

This work is an extension of my thesis on dynamic network flows. I again thank my thesis supervisor, Arthur F. Veinott, Jr., for all his help and encouragement. I also thank Paul Zipkin and the anonymous referees whose suggestions led to numerous improvements in the exposition.

REFERENCES

BARTLETT, T. E. 1957. An Algorithm for the Minimum Number of Transport Units to Maintain a Fixed Schedule. *Naval Res. Logist. Quart.* **4**, 139–149.

BARTLETT, T. E., AND A. CHARNES. 1957. Cyclic Scheduling and Combinatorial Topology: Assignment and Routing of Motive Power to Meet Scheduling and Maintenance Requirements; II. Generalization and Analysis. *Naval Res. Logist. Quart.* **4**, 207–220.

CAMERON, K., AND J. EDMONDS. 1980. Coflow Polyhedra and Min-max Relations, presented at the Spring 1980 joint ORSA/TIMS conference in Washington, D.C.

DANTZIG, G. B. 1962. Consulting work with United Airlines.

DANTZIG, G. B., AND D. R. FULKERSON. 1954. Minimizing the Number of Tankers to Meet a Fixed Schedule. *Naval Res. Logist. Quart.* **1**, 217–222.

DILWORTH, R. P. 1950. A Decomposition Theorem for Partially Ordered Sets. *Ann. Math.* **51**, 161–166.

FORD, L. R., AND D. R. FULKERSON. 1962. *Flows in Networks*. Princeton University Press, Princeton, N.J.

FULKERSON, D. R. 1956. Note on Dilworth's Decomposition Theorem for Partially Ordered Sets. *Proc. Am. Math. Soc.* **7**, 701–702.

GAREY, M. R., D. S. JOHNSON, G. L. MILLER AND C. H. PAPADIMITRIOU. 1980.

The Complexity of Coloring Circular Arcs and Chords. *SIAM J. Algebraic Discrete Meth.* **1,** 216–227.

GOLUMBIC, M. 1980. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York.

LEKKERKERKER, C. B., AND J. CH. BOLAND. 1962. Representation of a Finite Graph by Intervals on a Real Line. *Fund. Math.* **51,** 45–64.

ORLIN, J. B. 1981a. Dynamic Network Flows, PhD. thesis, Department of Operations Research, Stanford University.

ORLIN, J. B. 1981b. The Complexity of Dynamic Languages and Dynamic Optimization Problems, in *13th Annual ACM Symposium on the Theory of Computing,* pp. 218–227.

ORLIN, J. B., M. A. BONUCCELLI AND D. P. BOVET. 1981. An $0(n^2)$ Algorithm for Coloring Proper Circular Arc Graphs. *SIAM J. Algebraic Discrete Meth.* **2.**

SIMPSON, R. W. 1969. Scheduling and Routing Models for Airline Systems, Report FTL-R68-3, Flight Transportation Lab, MIT, pp. 100–107 and 128–134.

TUCKER, A. 1975. Coloring a Family of Circular Arcs. *SIAM J. Appl. Math.* **29,** 493–502.

WOLLMER, R. D. 1980. An Airline Schedule Tail Routing Algorithm, presented at the Fall 1980 ORSA/TIMS conference in Colorado Springs.