Mathematical Programming 27 (1983) 214–231
North-Holland

MAXIMUM-THROUGHPUT DYNAMIC NETWORK FLOWS

James B. ORLIN
Sloan School of Management, M.I.T., Cambridge, MA 02139, USA

Received 9 July 1981
Revised manuscript received 7 June 1982

This paper presents and solves the maximum throughput dynamic network flow problem, an infinite horizon integer programming problem which involves network flows evolving over time. The model is a finite network in which the flow on each arc not only has an associated upper and lower bound but also an associated transit time. Flow is to be sent through the network in each period so as to satisfy the upper and lower bounds and conservation of flow at each node from some fixed period on. The objective is to maximize the throughput, the net flow circulating in the network in a given period, and this throughput is shown to be the same in each period. We demonstrate that among those flows with maximum throughput there is a flow which repeats every period. Moreover, a duality result shows the maximum throughput equals the minimum capacity of an appropriately defined cut.

A special case of the maximum dynamic network flow problem is the problem of minimizing the number of vehicles to meet a fixed periodic schedule. Moreover, the elegant solution derived by Ford and Fulkerson for the finite horizon maximum dynamic flow problem may be viewed as a special case of the infinite horizon maximum dynamic flow problem and the optimality of solutions which repeat every period.

Key words: Network Flow Algorithms, Integer Programming, Infinite Horizon Programming, Vehicle Scheduling, Periodic Scheduling.

1. Introduction

The model and problem formulation

We present and solve the maximum-throughput dynamic network flow problem, an infinite-horizon integer programming problem that involves flows evolving over time. In a given finite network, referred to henceforth as the 'static network', flow is to be sent in all of the arcs in each of an infinite number of periods. Each arc has an upper and a lower bound on the arc flow and an integral transit time, which is the number of periods that it takes for flow to pass through the arc. For a flow to be feasible it must satisfy all upper- and lower-bound constraints, and also satisfy conservation of flow at each node in each period except for the first few periods during which the flow is 'initialized'. (If we require that conservation of flow is satisfied in all periods, then the problem of determining a feasible flow becomes NP-hard as proved in Section 3.) For a given feasible dynamic flow, the throughput in period p is the net amount of flow in transit in that period, i.e. flow that has been initiated in some arc in period p or earlier but has not reached the head of the arc by the beginning of period p. In Section 2, we demonstrate that because there is conservation of flow, the throughput is the same for all periods except possibly the first few.

The maximum- (resp. minimum-) throughput dynamic network flow problem is to determine a feasible dynamic flow with maximum (resp. minimum) throughput. The maximum- and minimum-throughput problems are equivalent since each may be reduced to the other by multiplying the flows and bounds by -1. While this problem is apparently new, it has antecedents in the literature. For example, it is related to the maximum network flow problem. Both the dynamic max-throughput min-cut formula and the application of dynamic network flows to vehicle scheduling stem naturally from the ideas presented by Ford and Fulkerson [9].

The minimum-throughput dynamic network flow problem is closely related to the finite-horizon dynamic maximum-flow problem as presented and solved by Ford and Fulkerson [8]. As is discussed in detail in Section 4, the finite-horizon problem may be transformed into a special case of the minimum-throughput dynamic network-flow problem.

Finally, the maximum-throughput dynamic network flow problem is a specialization of the minimum convex-cost dynamic network flow problem which is presented and solved by Orlin [18]. The objective in the latter problem is to find a feasible integral flow with fixed throughput so as to minimize the long-run (Cesàro) average cost per period.

Optimality of stationary flows and the max-throughput min-cut theorem

We show that if there is a feasible bounded dynamic flow, then the supremum of the throughputs of all feasible dynamic flows equals that of all feasible dynamic flows that are *stationary*, i.e., the flow in each arc is the same in all periods. If we restrict attention to stationary flows, the maximum-throughput dynamic network-flow problem reduces to that of finding a maximum-profit feasible circulation in the static network with profit per unit of flow in an arc being its transit time. A stationary flow is obtained by repeating the static circulation in each period over the infinite horizon, and the throughput of the dynamic flow is the profit of the static circulation.

The optimality of stationary flows is proved as part of the main theoretical result in Section 2. Also proved is the Max-Throughput Min-Cut Theorem, which states that the maximum throughput of a feasible flow is the minimum capacity of a cut, where a cut is not defined in terms of the original static network, but rather in terms of an infinite 'dynamic network'; each node of the dynamic network is an ordered pair representing a node of the static network and a period of time. As part of the proof we construct a cut and a feasible stationary flow whose throughput is equal to the capacity of the cut.

An application: Minimizing the number of vehicles to meet a fixed periodic schedule

Consider a transportation firm (e.g. an airline) that *must* schedule vehicles (e.g. airplanes) each day over an infinite horizon so that certain routes are traveled at the same time each day. In addition, there are certain other routes that the firm *may* schedule, and deadheading is permitted. The objective is to determine a feasible schedule that minimizes the number of vehicles needed.

Dantzig [4], in consulting work for United Airlines, considered the above problem under the added restriction that a schedule is stationary, and he modeled the problem as the static version of the minimum-throughput dynamic network flow problem. By the previously mentioned results the induced stationary schedule he found is optimal over the class of all schedules. As Dantzig and his collaborators observed, the stationary flight schedules induce vehicle schedules that are periodic, but do not necessarily repeat daily.

2. The maximum throughput dynamic network flow problem

The static network and problem formulation

A *static network* is a quintuple $G=(N, A, t, l, u)$ defined on a graph with a set $N=\{1,\ldots,n\}$ of nodes and a set A of directed arcs, possibly containing loops (i.e., arcs joining a node to itself) and multiple arcs between two nodes. Associated with each arc a is a *transit time* t_a, which is the (possibly negative) integral number of periods that it takes for flow to travel from the tail of the arc to its head. Also associated with each arc a are (possibly $+\infty$) upper and (possibly $-\infty$) lower bounds u_a and l_a on the flow initiated therein in each period, with $u_a \geq l_a$. These networks have also been referred to in the literature as 'networks with transit times', for example by Lawler [14]. If a flow begins in one period in the tail of an arc with a negative transit time, then the flow arrives at the arc's head at an earlier period. This somewhat anomalous situation is interpreted in the vehicle-scheduling problem of Section 3 as airplanes that may cross the international date line and arrive the day before they left. Orlin [18] gives a further interpretation of negative transit times in an application to cyclic capacity scheduling.

If the tail and head of arc a are i and j respectively, then we may denote the arc as (i, j) in those cases in which no ambiguity will result. (There may be several arcs with the same tail and head.) For each node $i \in N$, let H_i (resp. T_i) be the set of arcs whose head (resp. tail) is i. Let $t_{\max}=\max_{a\in A}|t_a|$.

For each arc $a \in A$ and each period $p \geq 1$, let x_a^p be the flow in arc a initiating in period p and thus terminating in period $p+t_a$.

A *dynamic flow* $x=(x_a^p)$ is *feasible* if it satisfies the upper- and lower-bound constraints:

$$l_a \leq x_a^p \leq u_a \quad \text{for } a \in A, p=1,2,3,\ldots, \tag{2.1}$$

and satisfies *conservation-of-flow* constraints at each node after period $t_{\max}$, i.e.

$$\sum_{a\in T_i} x_a^p = \sum_{a \in H_i} x_a^{p-t_a} \quad \text{for } i \in N, p > t_{\max}. \tag{2.2}$$

Conservation of flow is not necessarily satisfied during the first $t_{\max}$ periods, which we may veiw as the intialization periods.

We define the flow *in transit* in arc a in period p with a feasible dynamic flow x to be

$$\sum_{j=p-t_a+1}^{p} x_a^j \quad \text{if } t_a \geqslant 1, \tag{2.3}$$

$$-\sum_{j=p+1}^{p-t_a} x_a^j \quad \text{if } t_a \leqslant -1, \tag{2.3$'$}$$

$$0 \qquad \text{if } t_a = 0. \tag{2.3$''$}$$

If $t_a \geqslant 1$, then the flow in transit in arc a in period p is the net amount of flow initiated in the tail of arc a in or prior to period p and arriving at the head of a subsequent to period p. If $t_a \leqslant -1$, then the flow in transit in arc a in period p is the negative of the net amount of flow initiated in the tail of arc a subsequent to period p and arriving at the head of a in or prior to period p.

To see that the sum (2.3′) is an appropriate analogue of (2.3), consider one unit of flow sent from the tail of arc a in period p and reaching the head of a in period $p+t_a$. With regard to the conservation-of-flow constraints this unit flow is equivalent to sending negative-one unit of flow from the head of a in period $p+t_a$ and reaching the tail of a in period p. In fact we can replace arc $a=(i,j)$ with arc $a=(j,i)$ such that $t_\alpha=-t_a$, $l_a=-u_a$, and $u_\alpha=-l_a$. The definition of flow in transit given in (2.3′) is consistent because one unit of flow in arc a should give the same contribution to the flow in transit as negative-one unit of flow in arc α.

Lemma 1. *The sum of the arc flows in transit in each period $p \geqslant t_{\max}$ of a feasible dynamic flow is equal to the sum of the flows in transit in period $t_{\max}$.*

Proof. Lemma 1 is a special case of Lemma 3, which we prove later in this section.

An intuitive justification of the above lemma is as follows. In order to preserve conservation of flow, all flow in transit after period $t_{\max}$ must be sent forward upon arriving at the head of an arc. Therefore, the total amount of flow in transit is constant after period $t_{\max}$.

We refer to the sum of the flows in transit in period $t_{\max}$ as the *throughput*. This important time-invariant parameter of dynamic network flows may be visualized as the amount of flow circulating in the network. Henceforth we denote the throughput of flow $x=(x_a^p)$ as f_x.

The *maximum-* (resp. *minimum-*) *throughput dynamic network flow problem* is to determine a feasible dynamic flow with maximum (resp. minimum) throughput. In this section we show that if there is a feasible bounded flow, then the supremum of the throughputs of all feasible flows is the same as that of all stationary feasible flows. Moreover, we define and prove a dynamic analog of the max-flow min-cut theorem of Ford and Fulkerson [7].

The dynamic network

Let $G = (N, A, t, l, u)$ be a static network. In order to express flows evolving over time as ordinary network flows expand G into an infinite network, called the *dynamic network*, and denote it by $G^\infty = (N^\infty, A^\infty, l^\infty, u^\infty)$ where $N^\infty = \{i^p : i \in N$ and $p \in \{1, 2, 3, \ldots\}\}$. Node $i^p \in N^\infty$ represents node i of N in period p. For each arc $a = (i, j) \in A$ and for $p \geq \max(1 - t_a, 1)$ there is an arc $a^p = (i^p, j^{p+t_a}) \in A^\infty$ corresponding to the flow x_a^p and representing the fact that flow may be sent from node i in period p through arc a and arrive at node j in period $p + t_a$. Furthermore, the lower and upper bounds for the flow in arc a^p are the same as those for arc a.

Figures 1 and 2 show a static network and the corresponding dynamic network. Here and in other diagrams the numbers on arcs refer to transit times.

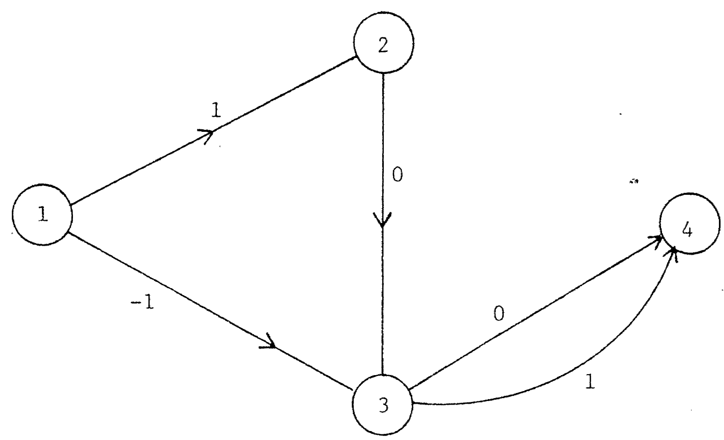

Fig. 1. A static network. The arc numbers are the transit times.

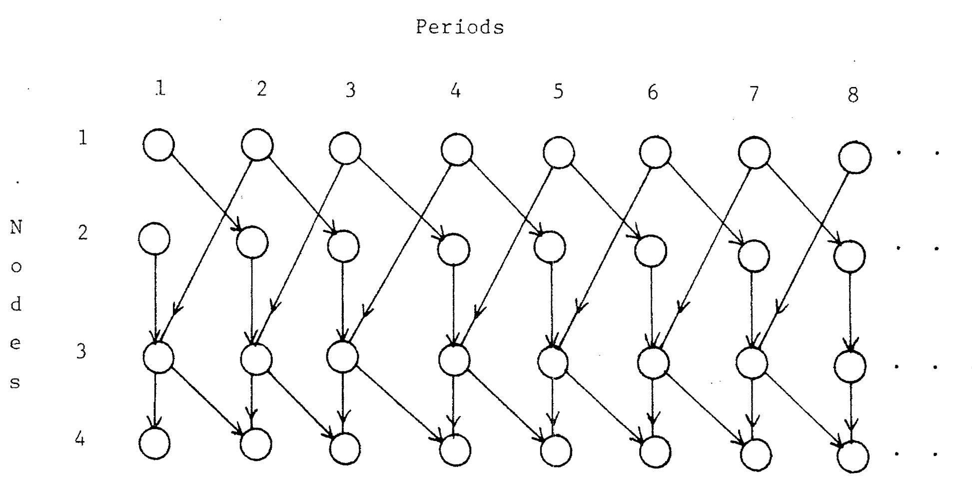

Fig. 2. The dynamic network derived by expanding the static network of Fig. 1.

The technique of expressing flows over time by expanding the network is now standard. It was used by Ford and Fulkerson [8] and by many others including Lawler [14].

Let $x = (x_a^p)$ be a feasible dynamic flow. Let us view x_a^p as the flow in arc a^p of A^∞. Then constraint (2.2) states that conservation of flow is satisfied at node i^p for $p > t_{\max}$. Thus a feasible dynamic flow is a circulation in the dynamic network except that conservation of flow is not necessarily satisfied at nodes i^p for $p \leq t_{\max}$.

Preliminaries: Paths, copies, and cuts

A *path* in a network (dynamic or not) is an alternating sequence of nodes and arcs $i_0, a_1, \ldots, a_k, i_k$ such that for each $j = 1, \ldots, k$ either a_j has head i_j and tail i_{j-1} or else it has head i_{j-1} and tail i_j. In the former case the arc is called a *forward arc* of the path; in the latter case it is called a *backward arc*. A path is called *directed* if every arc is a forward arc and *simple* if no node is repeated.

In a static network the *transit time* of a path is the sum of the transit times of the forward arcs of the path minus the sum of the transit times of the backward arcs.

A *cycle* is a path in which the initial node is the same as the final node. A cycle is *simple* if no node is repeated, except that the initial node and the final node are the same. For example, in the simple cycle $C = 1, (1, 2), 2, (2, 3), 3, (1, 3), 1$ of Fig. 1 the arc $(1, 3)$ is a backward arc, and the transit time of the cycle is 2. Had we oriented the cycle in the reverse direction the transit time would be -2.

A *static flow* $y = (y_a)$ is called *feasible* if it satisfies the upper and lower bound constraints (2.4) and is a circulation in the static network G, i.e. satisfies (2.5).

$$l_a \leq y_a \leq u_a \quad \text{for } a \in A \tag{2.4}$$

and

$$\sum_{a \in H_i} y_a - \sum_{a \in T_i} y_a = 0 \quad \text{for } i \in N. \tag{2.5}$$

If $a \in A$ and $p \geq 1$, then arc $a^p \in A^\infty$ will be called the pth *copy* of arc a, or simply a *copy* of a. The pth copy of arc a is not defined for $p \leq -t_a$. Similarly i^p is called the pth *copy* of node i. Let $P = i_0, a_1, \ldots, a_k, i_k$ be a path in G. The the pth *copy* of P is the path P', if one exists, in G^∞ with k arcs such that the first node is i^p and such that if a_j is a forward (resp. backward) arc of P then the jth arc of P is that copy of a_j whose tail (resp. head) is the jth node of P. To illustrate this concept, consider the simple path $P = 3, (1, 3), 1, (1, 2), 2$ in Fig. 1. Then the pth copy of P is the path $3^p, (1^{p+1}, 3^p), 1^{p+1}, (1^{p+1}, 2^{p+2}), 2^{p+2}$ in Fig. 2.

Lemma 2. *Let C be a simple cycle of transit time $t \geq 1$ in the static network. Then the infinite number of copies of C in the dynamic network comprise t node-disjoint infinite paths therein.*

Proof. First, ignore the finite number of copies that are not defined. Let the kth copy be the first copy that is defined, and let i^k be the first node of this path. For each $p \geq k$ it is easily verified that the pth copy of C is a path from i^p to i^{p+t}. If we concatenate the pth copies for $p = k, k+t, k+2t, k+3t, \ldots,$ then we obtain the first infinite path. We obtain $t-1$ additional paths with initial nodes $i^{k+1}, \ldots, i^{k+t-1}$ in the same manner. These paths are node-disjoint since for $j \neq i$, the node j^p may appear in at most one of the copies of C. □

As an example of the above lemma, consider the cycle $C = 1, (1, 2), 2, (2, 3), 3, (1, 3), 1$ of Fig. 1. Then C has a transit time of 2. Moreover, the infinite number of copies of C partition into two node-disjoint infinite paths in the dynamic network of Fig. 2, as can be seen upon deleting all copies of node 4 and all incident edges.

A *cut* in G^∞ is a partition of the nodes of N^∞ into disjoint subsets S, $\bar{S}$ such that S is finite, and $i^p \in S$ for $i \in N$ and $p \leq t_{\max}$. This guarantees that conservation-of-flow is satisfied at each node of $\bar{S}$. The nodes of S are called the *source nodes* while the nodes of $\bar{S}$ are called the *sink nodes*. A cut $(S, \bar{S})$ is called *monotone* if $i^p \in \bar{S}$ implies that $i^{p+1} \in \bar{S}$ for all $i \in N$ and $p > t_{\max}$.

Given two disjoint sets S, T of nodes in N^∞, let $A(S, T)$ denote the subset of arcs in A^∞ with tail in S and head in T. The *upper capacity* of a cut $(S, \bar{S})$ is defined to be

$$\sum_{a^p \in A(S,\bar{S})} u_a - \sum_{a^p \in A(\bar{S},S)} l_a \tag{2.6a}$$

and may be interpreted as the maximum net flow from source nodes to sink nodes. The *lower capacity* of cut $(S, \bar{S})$ is defined to be

$$\sum_{a^p \in A(S,\bar{S})} l_a - \sum_{a^p \in A(\bar{S},S)} u_a \tag{2.6b}$$

and may be viewed as the minimum net flow from source nodes to sink nodes. The upper (resp. lower) capacity is defined to be $+\infty$ (resp. $-\infty$) whenever (2.6a) (resp. (2.6b)) involve any infinite numbers.

Figure 3 portrays a nonmonotone cut of the dynamic network of Fig. 2. The upper and lower bounds are given in Table 1, and the nodes of S are the white nodes of Fig. 3. The upper capacity of the cut is 4 while the lower capacity is -3.

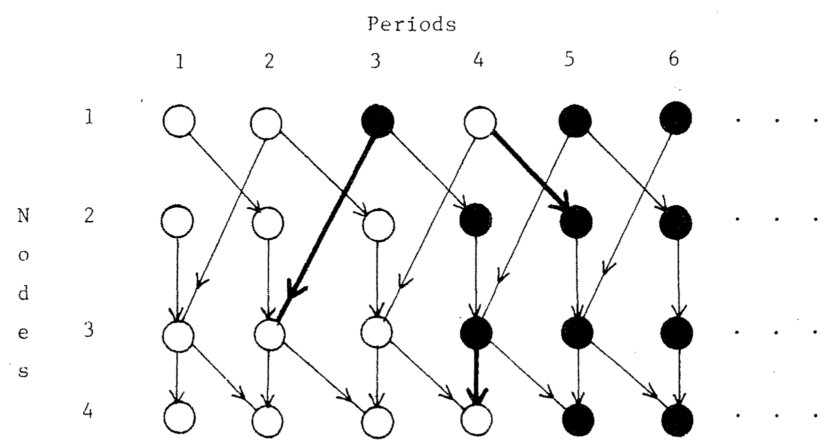

Fig. 3. A cut in the dynamic network of Fig. 2. The white nodes are in S. The arc $(1^4, 2^5)$ is in $A(S, \bar{S})$, while arcs $(1^3, 3^2)$ and $(3^4, 4^4)$ are in $A(\bar{S}, S)$.

Table 1
The parameters for the arcs of Fig. 1

arc	upper bound	lower bound	transit time
(1, 2)	3	0	1
(1, 3)	2	−1	−1
(2, 3)	2	0	0
(3, 4)	1	−1	0
(3, 4)	2	0	1

Max-throughput min-cut and the optimality of stationary flows

A *dynamic flow* $x=(x_a^p)$ is called *bounded* if its supremum norm is finite and *stationary* if $x_a^p=x_a^{p+1}$ for each $a\in A$ and $p=1,2,3,\ldots$. A feasible static flow $y=(y_a)$ *induces* a stationary dynamic flow $y^\infty=(x_a^p)$ with $x_a^p=y_a$ for $p=1,2,\ldots$ and $a\in A$. (In fact, there is a $1:1$ correspondence between feasible stationary flows in G^∞ and feasible static flows in G.) In this subsection we prove our main theoretical results. First, the upper (resp. lower) capacity of any cut in G^∞ provides an upper (resp. lower) bound on the throughput of a feasible flow. Second, if there is a feasible bounded flow, there is a cut in G^∞ whose upper (resp. lower) capacity equals the supremum (resp. infimum) of the throughputs of the stationary feasible dynamic flows.

Let x be a feasible dynamic flow and let $(S,\bar{S})$ be a cut in G^∞. We define the *flow across* $(S,\bar{S})$ to be

$$\sum_{a^p\in A(S,\bar{S})} x_a^p-\sum_{a^p\in A(\bar{S},S)} x_a^p.$$

Lemma 3. *Let x be a feasible dynamic flow and let $(S,\bar{S})$ be a cut. Then the throughput is equal to the flow across $(S,\bar{S})$ and so is bounded above by the upper capacity of the cut and bounded below by its lower capacity.*

Proof. It is clear that the flow across $(S,\bar{S})$ is bounded above by the upper capacity and bounded below by the lower capacity. It remains to show that the flow across $(S,\bar{S})$ is the throughput.

Consider first the case that $S=\{i^p\in N^\infty: p\le t_{\max}\}$. Then the flow across $(S,\bar{S})$ is exactly the flow in transit in period $t_{\max}$, which is the throughput.

We now prove the result inductively. Let $(S,\bar{S})$ be a cut. Let $S'=S-\{j\}$, where $j=i^r$ is any node of S for which $r>t_{\max}$. Then the flow across $(S,\bar{S})$ is equal to the flow across $(S',\bar{S}')$. To see this, note that the difference of the flows across the two cuts is

$$\sum_{a^p\in H_j} x_a^p-\sum_{a^p\in T_j} x_a^p$$

which is zero by conservation of flow.

The lemma now follows inductively, as we may start with any cut $(S, \bar{S})$ and progressively move one node at a time from S to $\bar{S}$ without altering the flow across the cut. Eventually we arrive at the set $S' = \{i^p \in N^\infty: p \leq t_{\max}\}$. □

Proof of Lemma 1. The new flow in transit in period q is the flow across $(S, \bar{S})$ where $S = \{i^p \in N^\infty: p \leq q\}$. Thus, Lemma 1 is a special case of Lemma 3. □

Theorem 4. *If there is a feasible bounded dynamic flow, then*

(1) (*Sufficiency of stationary flows*) *the supremum* (*resp. infimum*) *of the throughputs of all feasible dynamic flows equals that of all feasible stationary flows, with the last supremum* (*resp. infimum*) *being attained if it is finite.*

(2) (*Max-throughput min-cut*) *Moreover, the supremum* (*resp. infimum*) *of those throughputs equals the minimum* (*resp. maximum*) *upper* (*resp. lower*) *capacity of a cut, and this cut may be taken to be monotone.*

(3) (*Integrality*) *If also the upper and lower bounds are integral, then the stationary flows in* (1) *may be taken to be integral as well.*

Proof. First suppose that all upper and lower bounds on arc flows are finite. Consider the problem of finding a feasible static flow $y = (y_a)$ that maximizes

$$f_y = \sum_{a \in A} t_a y_a. \tag{2.7}$$

Furthermore, as is easily seen from (2.3) and (2.7), $f_y = f_{y^\infty}$, i.e. f_y is the throughput of the stationary flow y^∞ induced by the static flow y. The dual of the above static network flow problem is the linear program (2.8).

$$\text{Minimize} \quad \sum_{a \in A} \alpha_a u_a - \sum_{a \in A} \beta_a l_a \tag{2.8a}$$

subject to

$$\lambda_j - \lambda_i + \alpha_a - \beta_a = t_a \quad \text{for all } a = (i, j) \in A, \tag{2.8b}$$

$$\alpha_a, \beta_a \geq 0 \qquad \text{for } a \in A. \tag{2.8c}$$

Let $z^+ = \max(0, z)$, and $z^- = \max(0, -z)$. Since $u_a \geq l_a$ for each $a \in A$, we need consider only those solutions to (2.8) for which $\alpha_a = (t_a + \lambda_i - \lambda_j)^+$ and $\beta_a = (t_a + \lambda_i - \lambda_j)^-$. These solutions are feasible for any choice of λ.

Let (λ, α, β) be an integer-valued feasible solution to (2.8) and put $\lambda_{\min} = \min_i \lambda_i$. Let p^* be some integer greater than $t_{\max} - \lambda_{\min}$, and let $S = \{i^p \in N^\infty: p - \lambda_i \leq p^*\}$. Then $(S, \bar{S})$ is a monotone cut whose upper capacity is equal to the objective value of (2.8a) for (λ, α, β). To see this, suppose $a = (i, j)$. If $p - \lambda_i \leq p^*$ and $p - \lambda_j + t_a \geq p^* + 1$, then $a^p = (i^p, j^{p+t_a}) \in A(S, \bar{S})$, and there are $\alpha_a = (t_a + \lambda_i - \lambda_j)^+$ such copies of arc a. Similarly, $a^p \in A(\bar{S}, S)$ if $p - \lambda_i \geq p^* + 1$ and $p - \lambda_j + t_a \leq p^*$, and there are

$\beta_a = (t_a + \lambda_i - \lambda_j)^-$ such arcs. Thus the upper capacity of the cut is the objective value in (2.8a).

If there is a feasible static flow, then there is an optimal integral static flow y with objective value z and an integral optimal solution (λ, α, β) to (2.8) also with objective value z. The static flow y induces a feasible stationary flow y^∞ with throughput $f_y = z$ and solution (λ, α, β) induces a monotone cut $(S, \bar{S})$ with upper capacity equal to z. By Lemma 3, this stationary flow is optimal.

Consider next the case in which the bounds on some arcs may be infinite. If there is a sequence of feasible stationary flows with throughput unbounded from above, then by Lemma 3 each cut has infinite upper capacity. Suppose instead that there is no sequence of feasible *stationary* flows with throughput unbounded from above. Then since there is a feasible bounded dynamic flow $x = (x_a^p)$, there is a real number M that is a strict upper bound for both the absolute arc flows $|x_a^p|$ for all a and p and also for the absolute arc flows in each basic static flow (if any exist).

Consider next the static network $G' = (N, A, t, l', u')$ where $l'_a = \max(l_a, -M)$ and $u'_a = \min(u_a, M)$. The resulting dual program (2.8) is feasible and its objective value is bounded below by the throughput f_x by Lemma 3 and what was shown above. Hence, there is a maximum-profit basic static flow y and an optimal dual solution (λ, α, β) for (2.8). By complementary slackness, $u'_a = M$ implies $\alpha_a = 0$, and $l'_a = -M$ implies $\beta_a = 0$. Let $(S, \bar{S})$ be the monotone cut induced by (λ, α, β). Then no arc of $A(S, \bar{S})$ (resp., $A(\bar{S}, S)$) has an upper (resp. lower) bound equal to M (resp. $-M$). Hence, the capacity of the cut is unaltered if each arc bound of $\pm M$ is replaced by $\pm\infty$, and the throughput of the stationary flow y^∞ induced by y is the upper capacity of $(S, \bar{S})$ in G^∞ from what was shown above, completing the proof.

Finally, the result for the minimum-throughput problem is immediate from that for the maximum-throughput problem because the former reduces to the latter on replacing each (x_a^p, l_a, u_a) by $(-x_a^p, -u_a, -l_a)$. □

Example. Consider once again the static network described in Table 1. A minimum upper capacity cut can be obtained from Fig. 3 by shifting the node 1^4 from S to $\bar{S}$. The upper capacity of the resulting cut is 3. Moreover, we can produce a stationary dynamic network flow with a throughput of 3 as follows: send a flow of 1 unit in arcs $(1^p, 2^{p+1})$, $(2^p, 3^p)$, $(3^p, 4^{p+1})$ and a flow of -1 unit in arcs $(1^{p+1}, 3^p)$, $(3^p, 4^p)$ for each $p \geqslant 1$.

We note in passing that if x is a maximum-throughput dynamic network flow and $(S, \bar{S})$ is a minimum upper-capacity cut, then $x_a^p = u_a$ for $a^p \in A(S, \bar{S})$ and $x_a^p = l_a$ for $a^p \in A(\bar{S}, S)$.

Neither 1° nor 3° of Theorem 5 is true if we drop the restriction that there is a feasible bounded dynamic flow because there may be no feasible stationary flows in that event.

Example. (All feasible dynamic flows are unbounded.) Consider the static network described in Table 2. There is no feasible stationary flow, although there is a feasible

Table 2

arc	upper bound	lower bound	transit time
(1, 1)	∞	0	1
(2, 2)	0	$-\infty$	1
(1, 2)	1	1	0

dynamic flow $x = (x_a^p)$ given by

$$x_a^p = \begin{cases} -p, & a = (1, 1), \\ p + z, & a = (2, 2), \\ 1, & a = (1, 2), \end{cases}$$

where z is an arbitrary real number. Then $f_x = z$, so the supremum (resp. infimum) of the throughputs of these flows is $+\infty$ (resp. $-\infty$) and this is the upper (resp. lower) capacity of each cut.

On an alternate proof of (1) *of Theorem 5* (*Sufficiency of stationary flows*)

An alternate proof of (1) of Theorem 5 may be obtained by applying a result of Orlin [17] concerning the optimality of stationary solutions for dynamic convex programs. However, the resulting proof shows only that stationary flows suffice in the class of bounded feasible dynamic flows whereas the proof of Theorem 5 applies in the class of *all* feasible dynamic flows.

On the computational complexity of the maximum throughput problem

Theorem 5 implies that the maximum throughput problem can be solved as a static minimum cost network flow problem. One may ask whether this technique is really the fastest method in all cases. For example, if $t_{\max} = 1$ could we perhaps solve the problem as a finite maximum flow problem in order to speed up the computation? In fact, the computation time of the minimum cost static network flow problem and the maximum throughput problem are equivalent in the sense that a fast solution for one problem leads to a fast solution for the other problem. We see this equivalence as follows. First, we have previously seen that an optimal solution for the static problem induces an optimal stationary dynamic network flow. Suppose instead that $x = (x_a^p)$ is an optimal dynamic network flow. Let us also assume that we can express x efficiently as a flow with period q, i.e. $x_a^p = x_a^{p+q}$ for all $p \geqslant p^*$ (where p^* is sufficiently large) and where q is some positive integer. (We have assumed that to solve for x efficiently, we must be able to express the solution efficiently.) Then we can obtain an optimum static flow $y = (y_a)$ by letting

$$y_a = q^{-1}(x_a^p + x_a^{p+1} + \cdots + x_a^{p+q-1})$$

for $p = p^*$ and for all $a \in A$.

3. Everywhere-conservative feasible dynamic flows

In the maximum-throughput dynamic network-flow problem and again in the minimum convex-cost dynamic network-flow problem of Orlin [18], a feasible dynamic flow need not satisfy conservation of flow in the first few periods. This may be a significant relaxation of real-world constraints because conservation of flow usually has a physical interpretation. An interesting open question is how to 'phase into' an optimal stationary flow.

In many special instances it is easy to phase into an optimal stationary flow. In the vehicle scheduling problem of Section 5, it suffices to have all vehicles travel to their first departing site. In the cyclic capacity scheduling problem discussed in Orlin [18], it is possible to phase into any feasible stationary schedule within one day. However, in general, the 'phasing problem' is NP-hard, as is shown in this section. (The reader unfamiliar with NP-hardness should refer to Karp [13] or Garey and Johnson [11].)

A feasible dynamic network flow is called *everywhere conservative* if conservation of flow is satisfied in all periods including periods $1, \ldots, t_{\max}-1$. We refer to such flows as *conservative flows*, for brevity.

Theorem 5. *Determining whether there is a conservative flow is NP-hard.*

Proof. Consider the following version of the knapsack problem, which was proved NP-hard by Karp [13].

Knapsack problem. Given n distinct positive integers $d_1, \ldots, d_n$ and a positive integer b, do there exist nonnegative integers $w_1, \ldots, w_n$ such that $d_1 w_1 + \cdots + d_n w_n = b$?

Consider the two-node static network G described in Table 3. To prove the theorem we show that there is a conservative flow for G if and only if there is a feasible solution for the corresponding knapsack problem.

Table 3

arc	a_1	a_2	$\cdots$	a_n	α	β	γ
tail	1	1	$\cdots$	1	1	2	2
head	1	1	$\cdots$	1	1	1	1
lower bound	0	0	$\cdots$	0	-1	-1	1
upper bound	1	1	$\cdots$	1	-1	-1	1
transit time	d_1	d_2	$\cdots$	d_n	1	b	$b+1$

First, let G' be G with arcs α, β and γ deleted. Then there is a feasible solution for the knapsack problem if and only if there is a direct path in G' from node 1 to node 1 with transit time b. To complete the proof of the theorem, we show that there is a conservative flow for G if and only if there is a directed path in G' from node 1 to node 1 with transit time b.

Suppose x is a conservative flow. If we restrict x to copies of α, β and γ, the resulting feasible dynamic network flow satisfies conservation of flow at all nodes of the dynamic network except 1^1 which has a deficit of one unit and 1^{b+1} which has a surplus of one unit. Since x is a conservative flow, if we restrict x to copies of arcs $a_1, \ldots, a_n$ it must consist of a unit flow from 1^1 to 1^{b+1} and conserves flow elsewhere. The unit flow is sent along a path that is a copy of a path in G' *from* 1 to 1 with transit time b.

Conversely, if there is such a path in G', then there is a copy of the path in the dynamic network from 1^1 to 1^{b+1}. A conservative flow is created by sending 1 unit of flow along this path, 1 unit of flow in each copy of α and γ, and -1 unit of flow in each copy of β. □

The transformed dynamic network as restricted to copies of α, β and γ in the case that $b = 3$ is illustrated in Fig. 4.

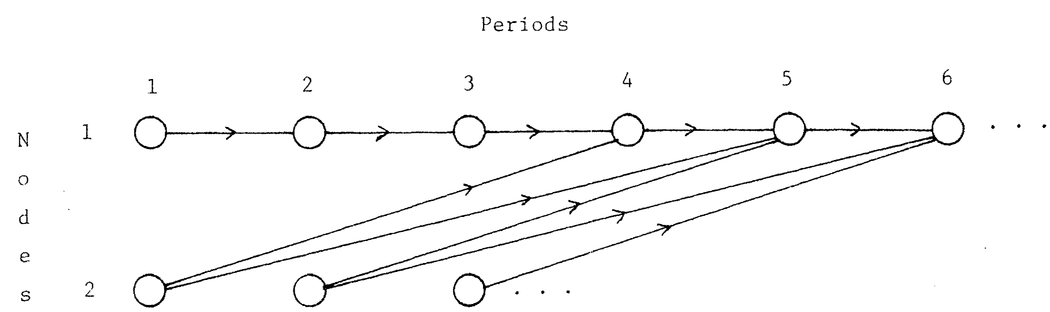

Fig. 4. The dynamic network as transformed from the knapsack problem with $b = 3$ as restricted to copies of α, β, γ.

The above construction also shows that the problem of determining a conservative flow is NP-hard even when the static network is restricted to having two nodes.

The transformation from the knapsack problem resulted in transit times that are possibly exponential in the number of arcs of the static network. It is an interesting open question whether the problem of determining a conservative flow for a network with polynomial bounded transit times is NP-hard.

4. Finite-horizon dynamic maximum flow

In this section we consider the q-period dynamic maximum-flow problem which Ford and Fulkerson [8] formulated and solved. The objective in this problem, which we shall henceforth call the *q-period problem*, is to find a maximum flow from a given source to a given sink in q periods where q is a given positive integer. Ford and Fulkerson showed that there is always an optimal flow (depending on q)

that is 'temporally repeated'. Such flows are essentially stationary. This elegant result unfortunately does not generalize to other finite-horizon problems such as the minimum cost dynamic flow problem or the universal dynamic flow problem as defined by Gale [10].

In this section we demonstrate that the q-period dynamic maximum flow problem may be transformed into a special case of the minimum-throughput dynamic network flow problem, which we will refer to in this section as the *infinite-horizon problem*. In this way, the Ford–Fulkerson results may be viewed as a specialization of the theory developed in Section 2 above.

The q-period problem

Let $G=(N, A, t, 0, u)$ be a static network in which each arc except one has a nonnegative transit time, a zero lower bound on its arc flow, and a positive integer-valued upper bound thereon. Let $N=\{1, \ldots, n\}$, and call node 1 the *source* and node n the *sink*. The remaining arc is a special arc $\alpha=(1, n)$ from source to sink with $t_\alpha=-q$, $l_\alpha=0$ and $u_\alpha=\infty$. This arc plays no role in the q-period problem but is important in the transformation of the q-period problem into the infinite horizon problem. Denote by $G^q=(N^q, A^q, t^q, 0, u^q)$ the q-period subnetwork of the dynamic network $G^\infty=(N^\infty, A^\infty, t^\infty, 0, u^\infty)$ induced by the nodes $i^p \in N^\infty$ for $i \in N$ and $1 \leq p \leq q$. Flow may be sent in G^q in periods $1, 2, \ldots, q$ for some fixed q. We call $x=(x_a^p)$ a *feasible q-period flow* in G^q if it satisfies (4.1), (4.2) and (4.3) below, viz. the upper- and lower-bound constraints

$$0 \leq x_a^p \leq u_a \quad \text{for } a \in A, p=1, \ldots, q, \tag{4.1}$$

the conservation-of-flow equations

$$\sum_{a \in T_i} x_a^p=\sum_{a \in H_i} x_a^{p-t_a} \quad \text{for } i=2, \ldots, n-1 \text{ and } p=1, \ldots, q, \tag{4.2}$$

and the flow prior to period 1 is zero, i.e., in the first q periods and at all nodes except for the source and sink,

$$x_a^p=0 \quad \text{for } a \in A \text{ and } p \leq 0. \tag{4.3}$$

We assume without loss of generality that $t_a<q$ for each arc $a \neq \alpha$. The *q-period problem* is to determine a feasible q-period flow that maximizes the amount of flow g_x that arrives at the sink where

$$g_x=\sum_{a \in H_n} \sum_{p=1}^{q-t_a} x_a^p. \tag{4.4}$$

Below we transform the q-period problem into a slightly modified version of the infinite horizon problem in which flow starts at period $-t_{\max}$ rather than period 0. Here we say that an infinite horizon flow (x_a^p) is *feasible* if it satisfies the bound constraints in periods $p=-t_{\max}, -t_{\max}+1, \ldots$ and satisfies conservations of flow at each node after period 0. (In particular, conservation of flow is satisfied at each node in G^q.) The *throughput* is the sum of the flows in transit in period 0. Finally, when we refer to the infinite network G^∞ we now refer to the infinite network induced by the node set $\{i^p: i \in N, p \geq -t_{\max}\}$.

Theorem 6. *Let y be a feasible integral dynamic network flow for the infinite horizon problem. Then y induces a feasible q-period flow x such that $g_x \geq -f_y$. Moreover, if y is a minimum throughput flow then $g_x = -f_y$.*

Proof. We first claim that if w is any minimum throughput flow for the infinite problem and if x is any maximum q-period flow then $g_x = -f_w$. This result is an immediate consequence of the optimality of stationary flows as stated in Theorem 4 and the optimality of 'temporally repeating flows' as proved by Ford and Fulkerson [8]. In particular, Ford and Fulkerson showed that the maximum flow from source to sink in the q-period problem is the negative of the minimum cost of the static network flow problem.

Thus, it now suffices to prove that y induces a q-period flow x with $g_x \geq -f_y$. We first let y' be the restriction of y to the finite network induced by the node set $\{i^p: i \in N, -t_{\max} \leq p \leq q + t_{\max}\}$. By flow decomposition theory y' may be expressed as the sum of unit flows along directed paths in addition to the sum of flows around directed circuits. Let $S = \{i^p: -t_{\max} \leq p \leq 0\}$ and let $T = \{i^p: q+1 \leq p \leq q + t_{\max}\}$. Let H be the collection of paths in the flow decomposition that originate at a node in T and terminate at a node in S. (If there is a flow of k units along a path, then this path is counted k times in H.) We now claim the following: (i) $|H| \geq -f_y$ and (ii) each path in H has as a subpath at least one path from source to sink in G^q.

To see that (i) is true, we first note that both the initial and terminal nodes of every path in the decomposition are in $S \cup T$ because conservation of flow in y' is satisfied at nodes in G^q. It follows that the throughput f_y is the number of paths from S to T in the decomposition minus the number of paths from T to S, and thus (i) is valid.

To see that (ii) is true, suppose that P is some path in H. Let 1^p be the first node in P such that $1^p \in G^q$ and no subsequent node in P is in T. (The node must be a copy of the source; else the predecessor of the node would be in G^q.) Let 1^r be the first node on the path P such that 1^r occurs subsequent to 1^p and $r < 0$. (Such a node must exist since there is a subpath originating at 1^p and ending in S.) Then the node n^{r+q} precedes 1^r on the path, and the subpath P' of P from 1^p to n^{r+q} is a source-sink path in G^q.

By (i) and (ii) above, the collection H of paths in the flow decomposition induce a feasible flow x in G^q such that $g_x \geq -f_y$, completing the proof. □

The q-period flows induced by stationary flows are precisely the 'temporally repeating' flows described by Ford and Fulkerson. Thus the reduction of the maximum q-period flow problem to the minimum throughput problem produces the same technique developed by Ford and Fulkerson.

Since minimum throughput flows induce maximum q-period flows, we might ask if maximum q-period flows can be extended into minimum throughput flows. In general, the answer is no as is illustrated by the example of Table 4 and Fig. 5. The 4 period flow of Fig. 5 cannot be extended into a feasible dynamic flow because the net amount of flow sent in arc $(4^5, 1^1)$ must be 2 in order to satisfy conservation of flow but there is at most one unit of flow that may enter node 4^5.

Table 4
A static network for the 4-period flow in Fig. 5

arc	lower bound	upper bound	transit time
(1, 2)	0	1	1
(1, 3)	0	1	1
(2, 3)	0	1	1
(3, 4)	0	1	1
(4, 1)	0	∞	−4

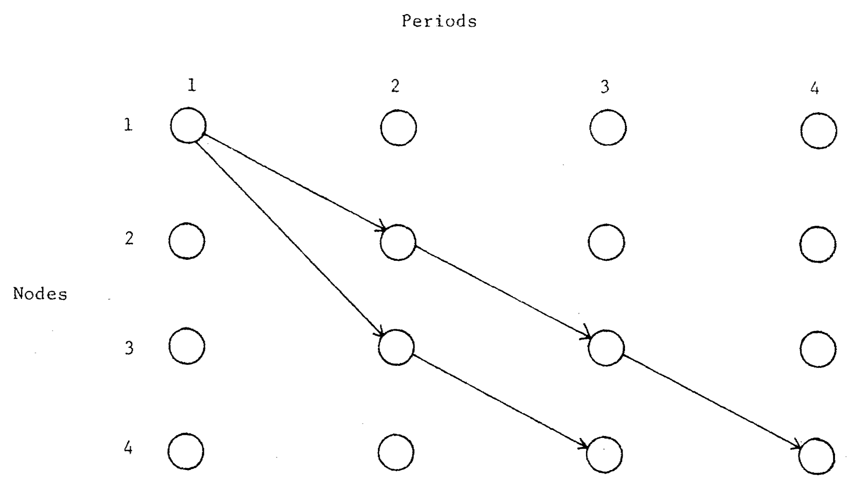

Fig. 5. A feasible 4-period flow that cannot be extended into a conservative infinite dynamic flow.

5. Minimizing the number of vehicles to meet a fixed periodic transportation schedule

Here we consider a routing problem that arises in the scheduling of vehicles for certain transportation industries, such as airlines and railroads. For convenience, we will borrow terminology from the airline industry.

The problem is to minimize the number of aircraft needed to meet a fixed schedule of daily repeating flights, each of which is either *required* or *optional*. The required flights must be flown daily, while the optional flights may be flown on any day at the scheduler's prerogative.

We assume that any plane may fly any route on any day. We do not make any a priori assumptions on the data, and we even allow the contingency that a flight may take several days. (This contingency is of little value is scheduling planes, but might be of value if we were to schedule trains.) Furthermore, we do not require that a feasible schedule be periodic, although our algorithm will always produce a periodic schedule.

Various versions of the above problem have been considered in the literature. Dantzig and Fulkerson [3] solved the problem of minimizing the number of vehicles to meet a fixed finite-horizon schedule. (It was Fulkerson's first paper on network flows.) The problems 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 [1] and the problem has been applied to railroad scheduling.

Dantzig [4] 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 is stationary. His technique, as described by Simpson [20], is equivalent to solving the static version of a corresponding minimum-throughput problem. By results of Section 2, the resulting optimal stationary solution that Dantzig obtains is optimal over the class of all feasible (possibly non-stationary) schedules.

More recently Wollmer [21] and independently Orlin [19] solved the special case of the airplane scheduling problem in which deadheading is allowed between any two cities. Wollmer's technique is to find an optimal solution for two consecutive days using the Dantzig–Ford [3] technique and then to show how to extend that schedule to an optimal schedule over an infinite horizon. Orlin's technique is to model the scheduling problem as a minimum chain-cover problem for 'periodic posets', and he solves the latter problem similarly to the technique presented in Section 2 by reducing it to a minimum cost flow problem.

Acknowledgment

I thank my thesis supervisor, Professor Arthur F. Veinott, Jr., for all his advice, encouragement, and help. He has made numerous suggestions that have led to improvements in both the content and exposition. I also thank Tom Magnanti and two anonymous referees for their many helpful suggestions.

References

[1] T.E. Bartlett, "An algorithm for the minimum number of transport units to maintain a fixed schedule", *Naval Research Logistics Quarterly* 4 (1957) 139–149.

[2] T.E. Bartlett and A. Charnes, "Cyclic scheduling and combinatorial topology: assignment and routing of motive power to meet scheduling and maintenance requirements. Part II: generalization and analysis", *Naval Research Logistics Quarterly* 4 (1957) 207–220.

[3] G.B. Dantzig and D.R. Fulkerson, "Minimizing the number of tankers to meet a fixed schedule", *Naval Research Logistics Quarterly* 1 (1954) 217–222.

[4] G.B. Dantzig, Consulting work for United Airlines (1962).

[5] J. Edmonds and R.M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems", *Journal of the Association of Computing Machinery* 19 (1972) 248–264.

[6] J. Folkman and D.R. Fulkerson, "Flows in infinite graphs", *Journal of Combinatorial Theory* 8 (1970) 30–44.

[7] L.R. Ford and D.R. Fulkerson, "Maximum flow through a network", *Canadian Journal Mathematics* 8 (1956) 399–404.

[8] L.R. Ford and D.R. Fulkerson, "Constructing maximal dynamic flows from static flows", *Operations Research* 6 (1958) 419–433.

[9] L.R. Ford and D.R. Fulkerson, *Flows in networks* (Princeton University Press, Princeton, NJ, 1962).

[10] D. Gale, "Transient flows in networks", *Michigan Mathematical Journal* 6 (1959) 59–63.

[11] M.R. Garey and D.S. Johnson, *Computers and intractibility: a guide to the theory of NP-completeness* (W.H. Freeman and Co., San Francisco, 1979).

[12] F. Glover, D. Klingman and C. McMillan, "The netform concept: a more effective model form and solution procedure for large scale non-linear problems", Management Science Report No. 77-4, Graduate School of Business Administration, University of Colorado (Boulder, CO, 1977).

[13] R.M. Karp, "Reducibility among combinatorial problems", in: R.E. Miller and J.W. Thatcher, eds., *Complexity of Computer Computations* (Plenum Press, New York, 1972) pp. 85–103.

[14] E.L. Lawler, *Combinatorial optimization: networks and matroids* (Holt, Rinehart and Winston, New York, 1976).

[15] W.L. Maxwell and R.C. Wilson, "Analysis of dynamic material handling systems by network flow", Working Paper No. 4, Department of Industrial Operations Engineering, University of Michigan (Ann Arbor, Michigan, 1978).

[16] E. Minieka, "Maximal lexicographic and dynamic network flows", *Operations Research* 21 (1973) 517–527.

[17] J.B. Orlin, "Dynamic convex programming," Chapter II, Ph.D. dissertation, Department of Operations Research, Stanford University (Stanford, CA, 1981).

[18] J.B. Orlin, "Minimum-convex-cost dynamic network flows, Chapter IV", Ph.D. dissertation, Department of Operations Research, Stanford University (Stanford, CA, 1981).

[19] J.B. Orlin, "Minimizing the number of vehicles to meet a fixed periodic schedule: An application of periodic posets", *Operations Research* 30 (1982) 760–776.

[20] R.W. Simpson, "Scheduling and routing models for airline systems", Report FTL-R68-3, Department of Aeronautics and Astronautics, M.I.T. (Cambridge, MA, 1969) pp. 100–107 and 128–134.

[21] R.D. Wollmer, "An airline schedule tail routing algorithm", presented at the ORSA/TIMS Conference (Colorado Springs, CO, 1980).