# CONSTRUCTING MAXIMAL DYNAMIC FLOWS FROM STATIC FLOWS

**L. R. Ford, Jr., and D. R. Fulkerson**

*The Rand Corporation, Santa Monica, California*

(Received January 6, 1958)

A network, in which two integers $t_{ij}$ (the traversal time) and $c_{ij}$ (the capacity) are associated with each arc $P_iP_j$, is considered with respect to the following question What is the maximal amount of goods that can be transported from one node to another in a given number $T$ of time periods, and how does one ship in order to achieve this maximum? A computationally efficient algorithm for solving this dynamic linear-programming problem is presented The algorithm has the following features (*a*) The only arithmetic operations required are addition and subtraction (*b*) In solving for a given time period $T$, optimal solutions for all lesser time periods are a by-product (*c*) The constructed optimal solution for a given $T$ is presented as a relatively small number of activities (chain-flows) which are repeated over and over until the end of the $T$ periods Hence, in particular, hold-overs at intermediate nodes are not required (*d*) Arcs which serve as bottlenecks for the flow are singled out, as well as the time periods in which they act as such (*e*) In solving the problem for successive values of $T$, stabilization on a set of chain-flows (*see* (*c*) above) eventually occurs, and an a priori bound on when stabilization occurs can be established The fact that there exist solutions to this problem which have the simple form described in (*c*) is remarkable, since other dynamic linear-programming problems that have been studied do not enjoy this property

SUPPOSE there is given a network in which each arc has associated with it two positive integers, one a commodity-flow capacity, the other a commodity-traversal time, and assume that commodity flow originates at some particular node (the *source*) and is ultimately destined for some other node (the *sink*) If each remaining node of the network can either transship the commodity immediately after receiving it or hold it for later shipment, what is the maximal amount that can be shipped from source to sink in any given number of time periods?

Section I contains a mathematical formulation of the maximal dynamic-flow problem described above and an interpretation of the problem as a static problem in an expanded network An algorithm for solving the problem is presented in Sec II The construction of a maximal dynamic flow for $T$ time periods proceeds in two stages first, a static flow is produced that maximizes a certain linear functional (Routine I, Theorem 3), second, this flow is decomposed into chain flows from source to sink (Routine II) A maximal dynamic flow can then be generated from the decomposition by starting each chain flow at time zero and continuing each as long as there is enough time left in the $T$ periods for the flow along the chain to arrive at the sink. Some preliminary theorems are given in Sec III and a few observations concerning the algorithm follow in Sec IV Section V presents a short proof that a dynamic flow obtained from the algorithm is maximal The concluding section illustrates the computation with a numerical example

Routine I of the algorithm, which constructs a static flow, is essentially a primal-dual method[5] for a capacitated transshipment problem An interesting feature of the method is that the construction of the desired static flow for $T$ periods proceeds from the one previously generated for $T-1$ periods This, coupled with the fact that eventually stabilization on a maximal static flow occurs, makes the solution of the dynamic problem for all $T$ an easy task

We have tried to keep the presentation reasonably self-contained However, it is likely that background material on static network-flow problems[1–5] would aid the reader, even though an attempt has been made to state the pertinent definitions and concepts

## I. FORMULATION OF PROBLEM

LET $S$ be the given network, consisting of nodes, $P_0, P_1, \ldots, P_n$, and oriented arcs joining certain pairs of nodes, the arc from $P_i$ to $P_j$ being denoted by $P_iP_j$, assume $P_0$ is the source, $P_n$ the sink Each arc $P_iP_j$ has a *capacity* $c_{ij}$ and a *traversal time* $t_{ij}$, each of these being a positive integer Finally, a positive integer $T$, the total number of time periods in question, is specified

Let $y_{ij}(t)$, for $t=0, 1, \ldots, T-t_{ij}$, represent the amount of flow in arc $P_iP_j$ that leaves $P_i$ at time $t$ (and will consequently arrive at $P_j$ at time $t+t_{ij}$), let $v(T)$ represent the total amount of flow that leaves the source (and hence also the total amount that enters the sink) during the $T$ periods $(0, 1), (1, 2), \cdots, (T-1, T)$, and let $y_{ii}(t)$ represent the holdover at node $P_i$ from time $t$ to $t+1$ Then the maximal dynamic-flow problem may be stated as the linear program

$$\text{maximize } v(T) \tag{1}$$

subject to the constraints

$$\begin{aligned} &\textstyle\sum_{j=1}^{j=n}\sum_{t=0}^{t=T}[y_{0j}(t)-y_{j0}(t-t_{j0})]-v(T)=0, && \\ &\qquad\textstyle\sum_{j=0}^{j=n}[y_{ij}(t)-y_{ji}(t-t_{ji})] \qquad =0, && (i=1,\cdots,n-1,\ t=0,\cdots,T) \\ &\textstyle\sum_{j=0}^{n-1}\sum_{t=0}^{t=T}[y_{nj}(t)-y_{jn}(t-t_{jn})]+v(T)=0, && \\ &\qquad\qquad 0\leqq y_{ij}(t)\leqq c_{ij}, && \end{aligned} \tag{2}$$

where $t_{ii}=1$, $c_{ii}=\infty$ [It is also assumed in (2), and elsewhere, that a variable $y_{ij}(t)$ $(i\neq j)$ is suppressed if either $t<0$ or $P_iP_j$ is not an arc of $S$ ] If $\{y_{ij}(t)\}$ and $v(T)$ satisfy (2), we shall refer to $\{y_{ij}(t)\}$ as a *dynamic flow* in $S$ for $T$ periods, and call $v(T)$ its *value*, if also $v(T)$ is maximal, then $\{y_{ij}(t)\}$ is a *maximal dynamic flow* in $S$

A *static flow* $\{x_{ij}\}$ in $S$ of *value* $v$ satisfies

$$
\begin{aligned}
&\textstyle\sum_{j=1}^{j=n} (x_{0j}-x_{j0})-v =0, \\
&\textstyle\sum_{j=0}^{j=n} (x_{ij}-x_{ji}) \quad =0, \quad (i=1,\cdots,n-1) \quad (3)\\
&\textstyle\sum_{j=0}^{n-1} (x_{nj}-x_{jn})+v=0, \\
&\qquad 0\leqq x_{ij}\leqq c_{ij},
\end{aligned}
$$

where, as before, $x_{ij}$ appears only if $P_iP_j$ is an arc of $S$ A *maximal static flow* is a static flow of maximal value

A network $D(t)$ that represents (1) and (2) as a static maximal-flow problem may be constructed as follows Let $P_i^t$ $(i=0, \cdots, n, t=0, \cdots, T)$ be the nodes of $D(T)$, and define oriented arcs $P_i^tP_i^{t+1}$ $(0\leqq t\leqq T-1)$ with capacities $\infty$, and, for $i\neq j$, $P_i^tP_j^{t+t_{ij}}$ $(0\leqq t\leqq T-t_{ij})$, with capacities $c_{ij}$ corresponding to arcs $P_iP_j$ of $S$ The sources in $D(T)$ are $P_0^t$ and the sinks are $P_n^t$, for all $t$ * Thus, if $y_{ij}(t)$ represents the amount of flow in arc $P_i^tP_j^{t+t_{ij}}$, relations (2) are the conditions for a static flow in $D(T)$ of value $v(T)$

Since the algorithm for constructing a maximal dynamic flow in $S$ for $T$ periods [or a maximal static flow in $D(T)$] deals only with static flows in $S$, we shall make no use of the network $D(T)$ except in the maximality proof of Sec 5

## II. THE ALGORITHM

ROUTINE I of the algorithm is an iterative process that has as final output an integral static flow $\{x_{ij}\}$, together with a set of integers $\{\pi_i\}$, one for each node $P_i$, such that

$$\pi_0=0, \quad \pi_n=T+1, \quad \pi_i\geqq 0, \tag{4a}$$

$$\pi_i+t_{ij}>\pi_j\rightarrow x_{ij}=0, \tag{4b}$$

$$\pi_i+t_{ij}<\pi_j\rightarrow x_{ij}=c_{ij} \tag{4c}$$

To state the routine, we suppose that we have an integral flow $\{x_{ij}\}$ and node integers $\{\pi_i\}$ satisfying (4) with $\pi_n=t$, and wish to construct $\{x'_{ij}\}$ and $\{\pi'_i\}$ satisfying (4) with $\pi'_n=t+1$. To start, we might take all $x_{ij}=0$ and all $\pi_i=0$

* Strictly speaking, we should have a single source and a single sink This can be accomplished, if desired, by joining all sources in $D(T)$ to a new source by arcs of infinite capacity, and similarly for the sinks

Arcs $P_iP_j$ for which $\pi_i+t_{ij}=\pi_j$ will be called *admissible* arcs Note that at most one member of the pair $P_iP_j$, $P_jP_i$ will be admissible, and that initially no arcs are admissible During the routine a node is either *unlabeled*, or *labeled but unscanned*, or *labeled and scanned* Initially all nodes are unlabeled

### Routine I

(*a*) To $P_0$ assign the label $[P_n^+, \infty]$, consider $P_0$ as unscanned

(*b*) Take any labeled, unscanned node $P_i$ (initially $P_0$ will be the only such node), suppose it is labeled $(P_k^\pm, h)$ To all nodes $P_j$ that are unlabeled and such that $P_iP_j$ is admissible and $x_{ij}<c_{ij}$, assign the label $[P_i^+, \min(h, c_{ij}-x_{ij})]$ To all nodes $P_j$ that are now unlabeled and such that $P_jP_i$ is admissible and $x_{ji}>0$, assign the label $[P_i^-, \min(h, x_{ji})]$ Consider $P_i$ as scanned and the newly labeled $P_j$ (if any) as unscanned Repeat until $P_n$ is labeled or until no new labels are possible and $P_n$ is unlabeled In the former case go to Routine I(*c*) below, in the latter case, let the present flow be denoted by the new name $\{x'_{ij}\}$ and proceed to Routine I(*d*).

(*c*) If $P_n$ is labeled $[P_k^+, h]$, replace $x_{kn}$ by $x_{kn}+h$, if $P_n$ is labeled $[P_k^-, h]$,* replace $x_{nk}$ by $x_{nk}-h$ In either case, next turn attention to $P_k$ In general, if $P_k$ is labeled $[P_j^+, m]$, replace $x_{jk}$ by $x_{jk}+h$, and if labeled $[P_j^-, m]$, replace $x_{kj}$ by $x_{kj}-h$, in either case turning attention then to $P_j$ Ultimately the node $P_0$ is reached, at this point stop the replacement process † Starting with the new flow‡ thus generated, discard the old labels and repeat Routines I(*a*) and I(*b*) until the latter case of I(*b*) obtains

(*d*) Define $\pi'_i$ by

$$\pi'_i=\begin{cases}\pi_i & \text{if } P_i \text{ is labeled,}\\ \pi_i+1 & \text{if } P_i \text{ is unlabeled}\end{cases}$$

This ends Routine I

Repeat Routine I starting with $\{x'_{ij}\}$ and $\{\pi'_i\}$ (giving as new admissible links those $P_iP_j$ for which $\pi'_i+t_{ij}=\pi'_j$) and continuing until the value of $\pi_n$ has been increased to $T+1$ At this point a solution of the dynamic problem for $T$ periods is at hand by decomposing the final flow obtained into chain flows, and combining these chain flows as described in the introduction Routine II below uses a labeling procedure to effect this decomposition

### Routine II

(*a*) To $P_0$ assign the label $[P_n, \infty]$, consider $P_0$ as unscanned

(*b*) Take any labeled, unscanned node $P_i$, suppose it is labeled $[P_k, h]$ To all

---

* If one starts Routine I with a flow $\{x_{ij}\}$ such that $x_{nj}=0$, then $P_n$ will never receive a label of this form

† The replacement process alters the flow along a path from $P_n$ to $P_0$

‡ It is easily proved that the replacement yields a static flow in $S$ of which the value is $h$ units greater than the preceding flow value

nodes $P_j$ that are unlabeled and such that $x_{ij}>0$, assign the label $[P_i, \min(h, x_{ij})]$ Consider $P_i$ as scanned and the newly labeled $P_j$ (if any) as unscanned Repeat until either $P_n$ is labeled or new labels are impossible and $P_n$ is unlabeled In the former case, proceed to Routine II$(c)$, in the latter case, stop

$(c)$ If $P_n$ is labeled $[P_k, h]$, replace $x_{kn}$ by $x_{kn}-h$, next turning attention to $P_k$ In general, if $P_k$ is labeled $[P_j, m]$, replace $x_{jk}$ by $x_{jk}-h$, and proceed to $P_j$ Stop the replacement when $P_0$ is reached * This ends Routine II

Repeat Routine II until the latter case of Routine II$(b)$ obtains If Routine II is being applied to a static flow obtained by the methods described above, then when Routine II can no longer be repeated, every $x_{ij}$ will be zero

## III PRELIMINARY THEOREMS

IN THE theorems that follow, we shall use the notation $\{x_{ij}^t\}$, $\{\pi_i^t\}$ $(t=1, \quad, T+1)$ for the flows and corresponding node numbers produced sequentially by Routine I, i e , $\{x_{ij}^t\}$ and $\{\pi_i^t\}$ are the outputs of the $t$th application of Routine I Thus we see that all $x_{ij}^1=0$, $\pi_0^1=0$, and $\pi_i^1=1$ for $i>1$ (since initially no arcs are admissible, and the labeled set consists only of $P_0$) In general, we have $\pi_0^t=0$ and $\pi_n^t=t$

THEOREM 1 *The flow* $(x_{ij}^t)$ *and corresponding node numbers* $(\pi_i^t)$ *are integral and satisfy* (4) *with* $T+1=t$

*Proof* Since the theorem is valid for $\{x_{ij}^1\}$, $\{\pi_i^1\}$, we need only to prove it for $\{x_{ij}^{t+1}\}$, $\{\pi_i^{t+1}\}$ on the assumption it holds for $\{x_{ij}^t\}$, $\{\pi_i^t\}$

The integrality of the $x_{ij}^{t+1}$ follows from the assumption that the $x_{ij}^t$ and $c_{ij}$ are integers, so that the flow change $h$ of I$(c)$ is an integer (in fact, a positive integer, hence Routine I terminates)

The properties listed in (4a) and the integrality of the $\pi_i^{t+1}$ are obvious from the definition of I$(d)$ and the latter case of I$(b)$

For (4b), assume that $\pi_i^{t+1}+t_{ij}>\pi_j^{t+1}$ Then from I$(d)$ and the fact that we are dealing with integers, we have $\pi_i^t+t_{ij}\geqq\pi_j^t$ If strict inequality holds, then the arc $P_iP_j$ was inadmissible throughout the labeling procedure I$(a)$ and I$(b)$, and hence $x_{ij}^{t+1}=x_{ij}^t=0$ If, on the other hand, equality holds, then from I$(d)$ we have $\pi_i^{t+1}=\pi_i^t+1$, $\pi_j^{t+1}=\pi_j^t$, and hence $P_i$ is unlabeled and $P_j$ labeled at the conclusion of I$(b)$ Since $P_iP_j$ was admissible, we must have $x_{ij}^{t+1}=0$, as otherwise $P_i$ would be labeled from $P_j$, a contradiction Thus we have $x_{ij}^{t+1}=0$ in either case, proving (4b)

The proof of (4c) is similar

It will be convenient for later purposes to introduce arc numbers $\{\gamma_{ij}^t\}$ defined by

$$\gamma_{ij}^t=\max(0, \pi_j^t-\pi_i^t-t_{ij}) \tag{5}$$

* One sees that at the end of Routine II$(c)$ a chain carrying $h$ units of flow has been traced out (in reverse)

and to establish the following theorem·

THEOREM 2 *Let $P_{i_0}, P_{i_1}, \ldots, P_{i_k}$ ($i_0=0$, $i_k=n$) be any chain from $P_0$ to $P_n$ Then we have*

$$\sum_{l=0}^{k-1} (t_{i_l i_{l+1}}+\gamma^t_{i_l i_{l+1}}) \geqq t \tag{6}$$

*If, in addition, we have $x^t_{i_l i_{l+1}}>0$ for $l=0, 1, \ldots, k-1$, then $\gamma^t_{i_l i_{l+1}}$ is strictly positive for some $l$ and equality holds in* (6)

*Proof* The first assertion of the theorem results from summing the inequalities

$$\pi^t_i+t_{ij}+\gamma^t_{ij} \geqq \pi^t_j \tag{7}$$

along the chain and noting that $\pi^t_0=0$, $\pi^t_n=t$

If $x^t_{ij}>0$, then by (4b) equality holds in (7), hence equality holds in (6) if each arc of the chain has a positive amount of flow

For the remaining assertion of the theorem, assume that $x^t_{i_l i_{l+1}}>0$ for all $l$ Since $P_0$ is labeled and $P_n$ unlabeled, there is an $m$ with $P_{i_m}$ labeled and $P_{i_{m+1}}$ unlabeled If $x^{t-1}_{i_m i_{m+1}}>0$, then $\pi^{t-1}_{i_m}+t_{i_m i_{m+1}} \leqq \pi^{t-1}_{i_{m+1}}$ by (4b) If, on the other hand, $x^{t-1}_{i_m i_{m+1}}=0$, then since the flow in arc $P_{i_m i_{m+1}}$ has changed, the arc was admissible, or $\pi^{t-1}_{i_m}+t_{i_m i_{m+1}}=\pi^{t-1}_{i_{m+1}}$ Now by I($d$) we have $\pi^t_{i_m}=\pi^{t-1}_{i_m}$, $\pi^t_{i_{m+1}} = \pi^{t-1}_{i_{m+1}}+1$ Hence, since $\pi^{t-1}_{i_{m+1}}-\pi^{t-1}_{i_m}-t_{i_m i_{m+1}} \geqq 0$, we see that $\gamma^t_{i_m i_{m+1}}=\pi^t_{i_{m+1}} -\pi^t_{i_m}-t_{i_m i_{m+1}}>0$

THEOREM 3 *The flow $\{x^t_{ij}\}$ maximizes the linear form*

$$tv-\sum_{i,j} t_{ij}\, x_{ij} \tag{8}$$

*(where $v$ is the value of the flow $\{x_{ij}\}$) over all static flows in $S$ Moreover, the corresponding node and arc numbers $\{\pi^t_i\}$, $\{\gamma^t_{ij}\}$ solve the dual linear program.*

*Proof* The primal linear program considered is that of maximizing (8) subject to the constraints (3) The dual* of this problem is to minimize the form

$$\sum_{i,j} \gamma_{ij}\, c_{ij} \tag{9}$$

subject to the constraints

$$\begin{gathered} -\pi_0+\pi_n=t, \\ \pi_i-\pi_j+\gamma_{ij} \geqq -t_{ij}, \\ \gamma_{ij} \geqq 0 \end{gathered} \tag{10}$$

It is clear that the node and arc numbers satisfy (10) Thus, to prove the theorem, it suffices to establish the properties

$$\pi^t_i-\pi^t_j+\gamma^t_{ij} > -t_{ij} \to x^t_{ij}=0, \qquad \gamma^t_{ij}>0 \to x^t_{ij}=c_{ij} \tag{11}$$

---

* For a discussion of linear-programming duality theorems, *see*, for example, A J GOLDMAN AND A W TUCKER, "Theory of Linear Programming," *Linear Inequalities and Related Systems*, Ann Math Study 38, edited by H W Kuhn and A W Tucker, Princeton University Press, 1956

To see this, notice that for any flow $\{x_{ij}\}$ of value $v$ and numbers $\{\pi_i\}$, $\{\gamma_{ij}\}$ satisfying (10),

$$
\begin{aligned}
tv - \sum_{i,j} t_{ij}\, x_{ij} &\leqq (-\pi_0+\pi_n)\, v + \sum_{i,j} (\pi_i - \pi_j + \gamma_{ij})\, x_{ij} \\
&= (-\pi_0+\pi_n)\, v + \pi_0 \sum_j (x_{0j} - x_{j0}) + \sum_{i=1}^{n-1} \pi_i \sum_j (x_{ij} - x_{ji}) \\
&\qquad + \pi_n \sum_j (x_{nj} - x_{jn}) + \sum_{i,j} \gamma_{ij}\, x_{ij} \\
&\leqq \sum_{i,j} \gamma_{ij}\, c_{ij},
\end{aligned}
$$

with equality holding throughout if (and only if) $\pi_i - \pi_j + \gamma_{ij} > -t_{ij}$ implies $x_{ij} = 0$ and $\gamma_{ij} > 0$ implies $x_{ij} = c_{ij}$

Properties (11) are easy consequences of (4) and (5)

In particular, therefore, we have

$$tv^t - \sum_{i,j} t_{ij}\, x_{ij}^t = \sum \gamma_{ij}^t\, c_{ij}, \tag{12}$$

where $v^t$ is the value of the flow $\{x_{ij}^t\}$

Note that, for all sufficiently large $t$, the problem of maximizing (8) subject to (3) becomes that of finding a maximal static flow that minimizes $\sum_{i,j} t_{ij}\, x_{ij}$ over all maximal flows In order to make this statement more precise, we first state some definitions Consider a sequence of distinct nodes $P_{i_0}, P_{i_1}, \ldots, P_{i_k}$ ($i_0=0$, $i_k=n$), where, for each $l=0, 1, \ldots, k-1$, either $P_{i_l}P_{i_{l+1}}$ or $P_{i_{l+1}}P_{i_l}$ is an arc of $S$, singling out, for each $l$, one of these two possibilities, call the resulting collection of arcs a *path* in $S$ from $P_0$ to $P_n$ Thus a path differs from a chain by allowing the possibility of traversing an arc in the direction opposite to its orientation Let $\sum_{l=0}^{k-1} \pm t_{i_l i_{l+1}}$, the plus or minus sign being taken according as $P_{i_l}P_{i_{l+1}}$ or $P_{i_{l+1}i_l}$ was selected, be the *traversal time* of the path

THEOREM 4. *If $t$ is greater than the maximal path traversal time from $P_0$ to $P_n$, then the flow $\{x_{ij}^t\}$ is maximal and the sum $\sum_{i,j} t_{ij}\, x_{ij}$ is minimized by $\{x_{ij}^t\}$ over all maximal flows Hence Routine I stabilizes on this flow*

*Proof* If $\{x_{ij}^t\}$ is not maximal, it follows from the algorithm of reference 4* that there exists a flow $x_{ij}$ of value $v=v^t+h$, $h>0$, and a path from $P_0$ to $P_n$ such that

$$x_{ij} = \begin{cases} x_{ij}^t \pm h & \text{if } P_iP_j \text{ belongs to the path,} \\ x_{ij}^t & \text{otherwise,} \end{cases}$$

where the plus or minus sign is taken according as the arc $P_iP_j$ is traversed with or against its orientation Hence, letting $\tau$ be the traversal time of the path, we have

$$\sum_{i,j} t_{ij}(x_{ij} - x_{ij}^t) = \tau h < th = t(v - v^t),$$

or

$$tv^t - \sum_{i,j} t_{ij}\, x_{ij}^t < tv - \sum_{i,j} t_{ij}\, x_{ij},$$

* The algorithm of reference 4 consists of the labeling and replacement processes of Routine I, treating all arcs as admissible

which contradicts the first assertion of Theorem 3 Thus $\{x_{ij}^{t}\}$ is maximal, and consequently $\{x_{ij}^{t}\}$ minimizes $\sum_{i,j} t_{ij} x_{ij}$ over all maximal flows Since the flow is maximal, succeeding applications of Routine I do not change it

## IV. DISCUSSION

INASMUCH as Routine I can repeat flows before a maximal flow is reached, the question arises, in solving a dynamic problem for all $T$, of how to recognize the stage at which a maximal flow is attained This question can be answered in various ways For example, one can perform Routines I($a$) and I($b$), treating all arcs as admissible Then the flow is maximal if and only if the labeling stops with $P_n$ unlabeled (Lemma 2 of ref 4) Another way is just to repeat Routine I, since it can be shown that eventually a stage is reached where all arcs leading from labeled nodes to unlabeled nodes have positive arc numbers, and hence are saturated, whereas all arcs leading from unlabeled to labeled nodes are flowless The set of arcs leading from labeled to unlabeled nodes then constitutes a minimal cut* in $S$ and the flow is maximal

To obtain a maximal dynamic flow for $T$ time periods, the flow $\{x_{ij}^{T+1}\}$ is decomposed into chain flows by Routine II, and these are used to generate a dynamic flow as described in the next section We point out that an arbitrary flow can contain circularities or chain flows from $P_n$ to $P_0$ This cannot be the case for flows produced by Routine I, however, as is easily seen from Theorem 3 or directly from (4)

## V PROOF OF MAXIMALITY

LET $(P_{i_0}, P_{i_1}, \quad, P_{i_k}, h)$ be any one of the chain flows in a decomposition of $\{x_{ij}^{T+1}\}$, and define correspondents of this chain flow in $D(T)$, namely

$$(P_{i_0}^{t_0}, P_{i_1}^{t_1}, \quad, P_{i_k}^{t_k}, h), \qquad (i_0=0, i_k=n)$$

where

$$t_{l+1}=t_l+t_{i_l i_{l+1}}, \qquad (l=0, \quad, k-1)$$

and

$$0\leqq t_0, \qquad t_k\leqq T$$

This is to be done for all chain flows in the decomposition

That such chains exist in $D(T)$ follows from the second part of Theorem 2 by taking $t_0=0$ Then we have

$$t_k+\sum_{l=0}^{k-1} \gamma_{i_l i_{l+1}}^{T+1}=\sum_{l=0}^{k-1} (t_{i_l i_{l+1}}+\gamma_{i_l i_{l+1}}^{T+1})=T+1,$$

and since some arc number in this sum is positive, we have $t_k\leqq T$ From

* A cut in $S$ (separating $P_0$ from $P_n$) is a collection of arcs that meets every chain from $P_0$ to $P_n$, and that has no proper subset with this property It is easy to show that the sum of capacities of arcs forming a cut is an upper bound for flow values The max flow-min cut theorem[3] asserts the existence of a flow and a cut such that the value of the flow equals the cut capacity

the last displayed equation, we see in fact that the number of dynamic correspondents of this static chain flow is given by

$$T+1-\sum_{l=0}^{k-1} t_{i_l i_{l+1}}=\sum_{l=0}^{k-1} \gamma_{i_l i_{l+1}}$$

It is clear that the sum of all of the dynamic chain flows thus defined is a flow in $D(T)$, i e , if $y_{ij}(t)$ is defined as the sum of the numbers $h$ over all the dynamic chain flows that contain the arc $P_i^t P_j^{t+t_{ij}}$, then the constraints (2) are satisfied and we have

$$v(T)=\sum_r (T+1-\tau_r)\, h_r, \tag{13}$$

where $\tau_r$ is the traversal time of the $r$th chain in the decomposition and $h_r$ is the amount of flow along this chain It follows from (13) that

$$v(T)=(T+1)\, v^{T+1}-\sum_{i,j} t_{ij}\, x_{ij}^{T+1}, \tag{14}$$

and hence from (12) that

$$v(T)=\sum_{i,j} \gamma_{ij}^{T+1}\, c_{ij} \tag{15}$$

Now define the following set of arcs in $D(T)$

$$\Gamma=\{P_i^t P_j^{t+t_{ij}} \mid \pi_i^{T+1} \leqq t<\pi_j^{T+1}-t_{ij}\}, \tag{16}$$

i e , $\Gamma$ is the set of arcs that lead from any node of

$$N=\{P_i^t \mid \pi_i^{T+1} \leqq t\} \tag{17}$$

to its complement Since every source of $D(T)$ is in $N$, and every sink is in its complement because $\pi_0^{T+1}=0$, $\pi_n^{T+1}=T+1$, it follows that $\Gamma$ meets every chain of $D(T)$ from any source to any sink Thus the sum of capacities of arcs of $\Gamma$, which we denote by $c(\Gamma)$, is an upper bound for dynamic flow values But from (5), (16), and the fact that $\Gamma$ contains no arcs of the form $P_i^t P_i^{t+1}$, it follows that

$$c(\Gamma)=\sum_{i,j} \gamma_{ij}^{T+1}\, c_{ij} \tag{18}$$

Hence, from (15) and (18), the dynamic flow obtained from a decomposition of $\{x_{ij}^{T+1}\}$ is maximal, and $\Gamma$ is a minimal dynamic cut, i e , a minimal cut in $D(T)$ Thus we have established the following result

THEOREM 5 *The flow* $\{x_{ij}^{T+1}\}$ *generates a maximal dynamic flow for* $T$ *time periods, which has value* $v(T)$ *given by* (14) *or* (15) *Moreover, the set* $\Gamma$ *defined by* (16) *is a minimal dynamic cut*

It should perhaps be mentioned that the $c_{ij}$ were assumed to be integers only to ensure the termination of the labeling and replacement processes of Routine I The maximality proof of this section can be reworded to apply to any flow that maximizes $(T+1)\, v-\sum_{i,j} t_{ij}\, x_{ij}$, regardless of the integrality of the arc capacities

## VI. A NUMERICAL EXAMPLE

IN THE NETWORK of Fig 1, the first number on an arc is its capacity, the second its traversal time These are assumed to be symmetric $c_{ij}=c_{ji}$ and $t_{ij}=t_{ji}$. The complete solution of the example for all $T$, given in Table I, required less than an hour of hand computation

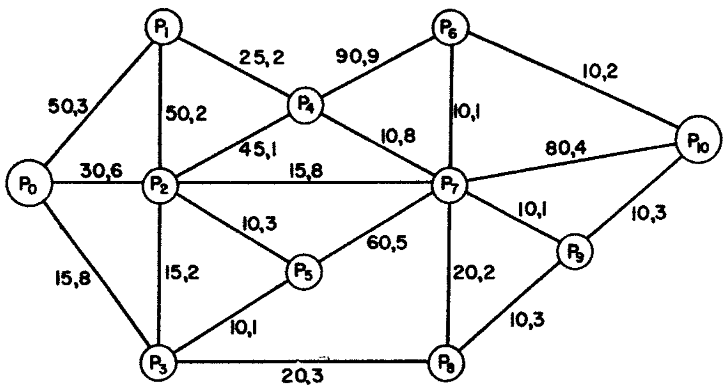


**Figure 1**

Figures 2–9 show the outputs of Routine I throughout the computation beginning with $\{x_{ij}^{15}\}=0$, $\{\pi_i^{15}\}$ as shown in Fig 2 From $\{\pi_i^{15}\}$, admissible arcs are determined (indicated by black arrowheads in Fig 2), and the labeling and replacement processes yield the flow $\{x_{ij}^{16}\}$ (shown as the numbers in the lower left-hand corner of the boxes in Fig. 2), together with the final labeled set of nodes for this cycle (the cross-hatched nodes of Fig 2). Then I(*d*) is applied to produce $\{\pi_i^{16}\}$ (shown in Fig 3), and Routine I

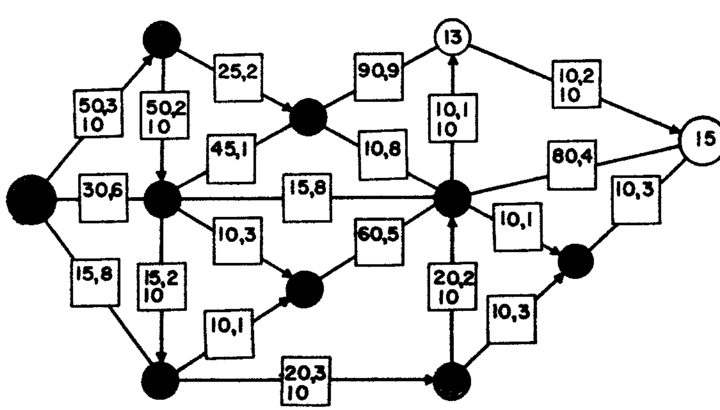


**Figure 2**

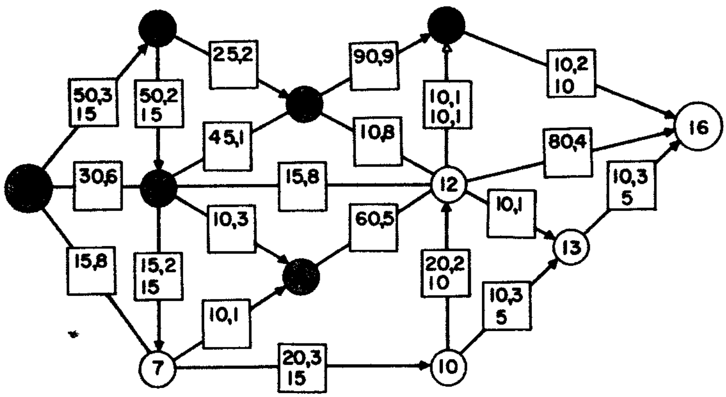


Figure 3

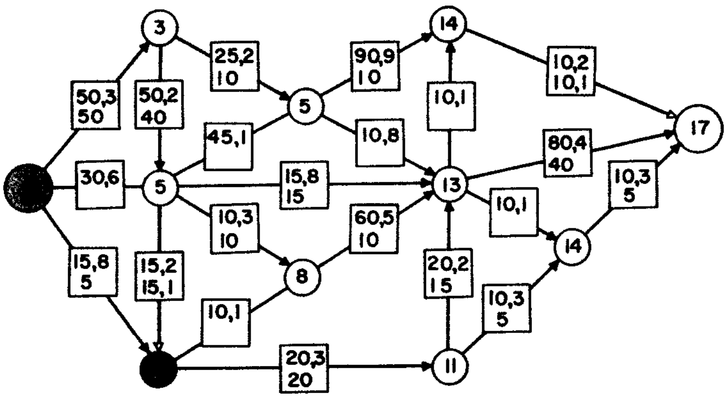


Figure 4

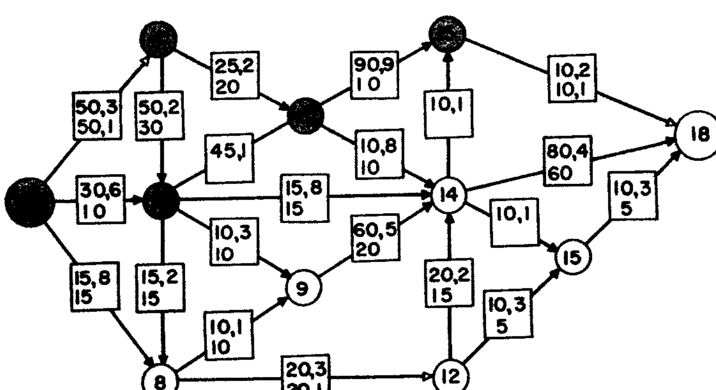


Figure 5

## TABLE I

Solution of Illustrative Numerical Example

| Total number of time intervals $T$ | Chain flow | Number of times chain used | Flow value | Minimal dynamic cut: Arc | Minimal dynamic cut: Time $t$ |
|---|---|---|---|---|---|
| 15 | $(P_0P_1P_2P_3P_8P_7P_6P_{10}, 10)$ | 1 | 10 | $P_7{}^tP_6{}^{t+1}$ | 12 |
| 16 | $(P_0P_1P_2P_3P_8P_7P_6P_{10}, 10)$ | 2 | 25 | $P_6{}^tP_{10}{}^{t+2}$ | 14 |
| | $(P_0P_1P_2P_3P_8P_9P_{10}, 5)$ | 1 | | $P_2{}^tP_3{}^{t+2}$ | 5 |
| 17 | $(P_0P_1P_2P_3P_8P_7P_{10}, 15)$ | 2 | 80 | $P_0{}^tP_1{}^{t+3}$ | 0 |
| | $(P_0P_1P_2P_7P_{10}, 15)$ | 1 | | $P_3{}^tP_8{}^{t+3}$ | 8 |
| | $(P_0P_1P_2P_5P_7P_{10}, 10)$ | 1 | | $P_6{}^tP_{10}{}^{t+2}$ | 15 |
| | $(P_0P_1P_4P_6P_{10}, 10)$ | 2 | | | |
| | $(P_0P_3P_8P_9P_{10}, 5)$ | 1 | | | |
| 18 | $(P_0P_1P_2P_3P_8P_7P_{10}, 15)$ | 3 | 155 | $P_0{}^tP_1{}^{t+3}$ | 0 |
| | $(P_0P_1P_2P_7P_{10}, 15)$ | 2 | | $P_0{}^tP_3{}^{t+8}$ | 0 |
| | $(P_0P_1P_4P_6P_{10}, 10)$ | 3 | | $P_2{}^tP_3{}^{t+2}$ | 6 |
| | $(P_0P_1P_4P_7P_{10}, 10)$ | 2 | | $P_2{}^tP_7{}^{t+8}$ | 6 |
| | $(P_0P_2P_5P_7P_{10}, 10)$ | 1 | | $P_2{}^tP_5{}^{t+3}$ | 6 |
| | $(P_0P_3P_5P_7P_{10}, 10)$ | 1 | | $P_3{}^tP_8{}^{t+3}$ | 9 |
| | $(P_0P_3P_8P_9P_{10}, 5)$ | 2 | | $P_4{}^tP_7{}^{t+8}$ | 6 |
| | | | | $P_6{}^tP_{10}{}^{t+2}$ | 15, 16 |
| 19 | $(P_0P_1P_2P_3P_8P_7P_{10}, 15)$ | 4 | 230 | $P_0{}^tP_1{}^{t+3}$ | 0 |
| | $(P_0P_1P_2P_7P_{10}, 15)$ | 3 | | $P_0{}^tP_3{}^{t+8}$ | 0, 1 |
| | $(P_0P_1P_4P_6P_{10}, 10)$ | 4 | | $P_2{}^tP_3{}^{t+2}$ | 6, 7 |
| | $(P_0P_1P_4P_7P_{10}, 10)$ | 3 | | $P_2{}^tP_5{}^{t+3}$ | 6, 7 |
| | $(P_0P_2P_5P_7P_{10}, 10)$ | 2 | | $P_2{}^tP_7{}^{t+8}$ | 6, 7 |
| | $(P_0P_3P_5P_7P_{10}, 10)$ | 2 | | $P_3{}^tP_8{}^{t+3}$ | 10 |
| | $(P_0P_3P_8P_9P_{10}, 5)$ | 3 | | $P_4{}^tP_7{}^{t+8}$ | 6, 7 |
| | | | | $P_6{}^tP_{10}{}^{t+2}$ | 15, 16, 17 |
| 20 | $(P_0P_1P_2P_3P_8P_7P_{10}, 15)$ | 5 | 310 | $P_0{}^tP_1{}^{t+3}$ | 0 |
| | $(P_0P_1P_2P_7P_{10}, 10)$ | 4 | | $P_0{}^tP_3{}^{t+8}$ | 0, 1, 2 |
| | $(P_0P_1P_4P_6P_{10}, 10)$ | 5 | | $P_2{}^tP_3{}^{t+2}$ | 6, 7, 8 |
| | $(P_0P_1P_4P_6P_7P_{10}, 5)$ | 2 | | $P_2{}^tP_5{}^{t+3}$ | 6, 7, 8 |
| | $(P_0P_1P_4P_7P_{10}, 10)$ | 4 | | $P_2{}^tP_7{}^{t+8}$ | 6, 7, 8 |
| | $(P_0P_2P_7P_{10}, 5)$ | 3 | | $P_1{}^tP_4{}^{t+2}$ | 4 |
| | $(P_0P_2P_5P_7P_{10}, 10)$ | 3 | | $P_3{}^tP_8{}^{t+3}$ | 11 |
| | $(P_0P_3P_5P_7P_{10}, 10)$ | 3 | | $P_4{}^tP_7{}^{t+8}$ | 7, 8 |
| | $(P_0P_3P_8P_9P_{10}, 5)$ | 4 | | $P_6{}^tP_{10}{}^{t+2}$ | 16, 17, 18 |

TABLE I—*Continued*

| Total number of time intervals $T$ | Chain flow | Number of times chain used | Flow value | Minimal dynamic cut: Arc | Minimal dynamic cut: Time $t$ |
|---|---|---|---|---|---|
| $21+k$ | $(P_0P_1P_2P_3P_8P_7P_{10}, 15)$ | $6+k$ | $395+85k$ | $P_0^tP_1^{t+3}$ | 0 |
| $(k=0, 1, 2,$ | $(P_0P_1P_2P_7P_{10}, 10)$ | $5+k$ | | $P_0P_3^{t+8}$ | $0,1, \quad ,3+k$ |
| $)$ | $(P_0P_1P_4P_6P_{10}, 10)$ | $6+k$ | | $P_2^tP_3^{t+2}$ | $6,7 \quad ,9+k$ |
| | $(P_0P_1P_4P_6P_7P_{10}, 5)$ | $3+k$ | | $P_2^tP_5^{t+3}$ | $6,7, \quad ,9+k$ |
| | $(P_0P_1P_4P_7P_{10}, 10)$ | $5+k$ | | $P_2^tP_7^{t+8}$ | $6,7, \quad ,9+k$ |
| | $(P_0P_2P_7P_{10}, 5)$ | $4+k$ | | $P_1^tP_4^{t+2}$ | 4 |
| | $(P_0P_2P_5P_7P_{10}, 10)$ | $4+k$ | | $P_3^tP_8^{t+3}$ | $12+k$ |
| | $(P_0P_3P_5P_7P_{10}, 10)$ | $4+k$ | | $P_4^tP_7^{t+8}$ | $7,8, \quad ,9+k$ |
| | $(P_0P_3P_8P_9P_{10}, 5)$ | $5+k$ | | $P_6^tP_{10}^{t+2}$ | $16, \quad ,19+k$ |
| | $(P_0P_2P_4P_6P_7P_{10}, 5)$ | $1+k$ | | $P_6^tP_7^{t+1}$ | $16, \quad ,16+k$ |

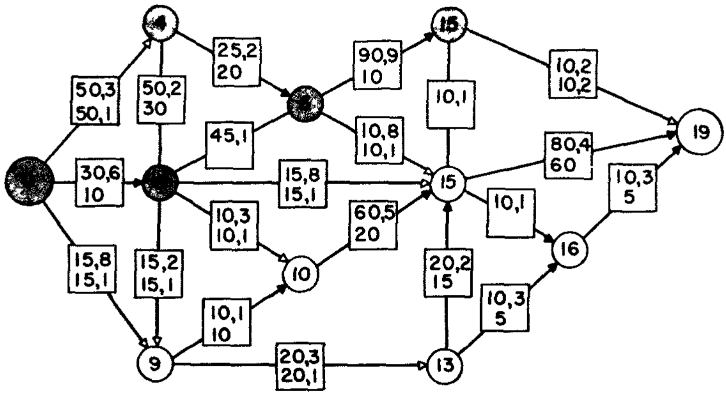


**Figure 6**

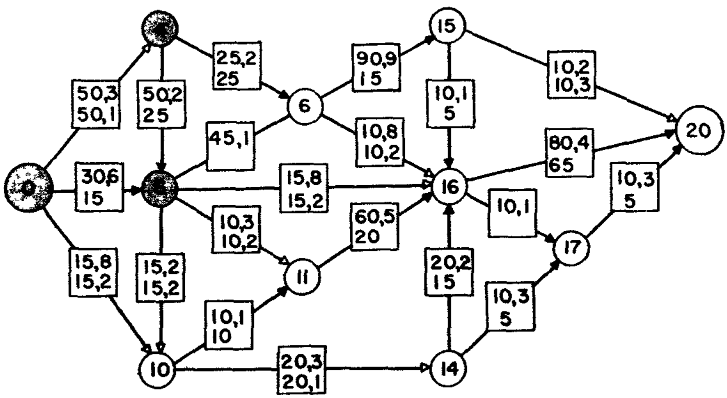


**Figure 7**

is then repeated with the new admissible arcs, giving the flow $\{x_{i,j}^{17}\}$ and labeled set (shown in Fig 3) White arrowheads indicate directions of flow in arcs that are inadmissible by virtue of having positive arc numbers, the latter being shown in the lower right-hand corner of the boxes (e g, arc $P_7P_6$ in Fig 3)

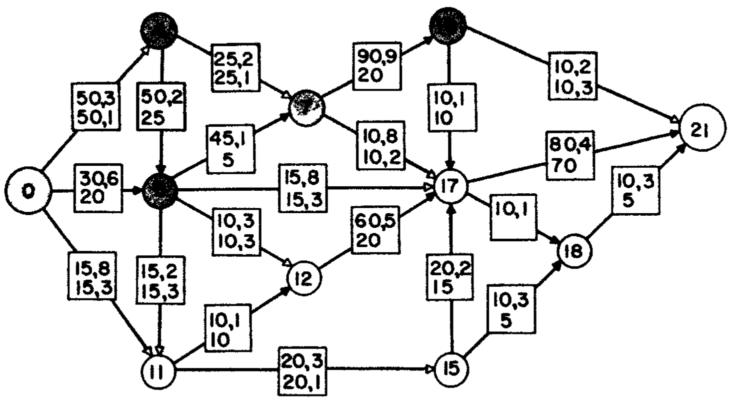


**Figure 8**

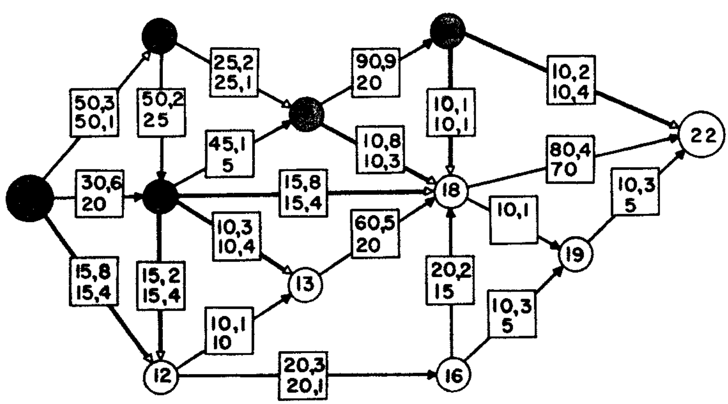


**Figure 9**

Routine I stabilizes on $\{x_{i,j}^{22}\}$ (shown in Figs 8 and 9), since the arcs leading from labeled to unlabeled nodes in Fig 9 all have positive arc numbers and the reverse arcs are flowless These arcs therefore constitute a minimal static cut in the network.

## REFERENCES

1 G B Dantzig and D R Fulkerson, "On the Min Cut Max Flow Theorem of Networks," pp 215–221 in H W Kuhn and A W Tucker, *Linear Inequalities*

*and Related Systems*, Annals of Mathematics Study No 38, Princeton University Press, Princeton, N J , 1956
2 L R FORD, JR ,"Network Flow Theory," The Rand Corporation Paper P-923, August, 1956
3 L R FORD, JR , AND D R FULKERSON, "Maximal Flow through a Network," *Canadian J Math* **8**, 399–404 (1956)
4 ———, "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," *Canadian J Math* **9**, 210–218 (1957)
5 ———, "A Primal Dual Algorithm for the Capacitated Hitchcock Problem " *Naval Res Log Quart* **4**, 47–54 (1957)