DISTRIBUTED ROUTING

by

ART O'LEARY

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE
DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 1981

Signature of Author ______________________
Department of Electrical Engineering and
Computer Science, January 13, 1981

Certified by ______________________
Robert G. Gallager
Thesis Supervisor

Accepted by ______________________
Arthur C. Smith, Chairman
Department Committee on Graduate Students

DISTRIBUTED ROUTING

by

ART O'LEARY

Submitted to the Department of Electrical Engineering
and Computer Science
on January 13, 1981 in partial fulfillment of the
requirements for the degree of Doctor of Philosophy.

ABSTRACT

In distributed routing each node receives some information about the network from its adjacent nodes and uses the information to determine the manner in which it forwards its traffic. This thesis gives three examples of distributed routing in a data communication network. A routing algorithm is then given where a generalized distributed routing procedure proposes a flow change and a central node determines the optimal scale of the proposed change. Small flows on long and unwanted paths are set to zero regardless of the scaling. The thesis shows that the iterative use of this algorithm converges to the optimal network cost, e.g. it minimizes the mean delay.

Thesis Supervisor: Robert G. Gallager

Title: Professor of Electrical Engineering
and Computer Science

Table of Contents

Chapter I.

Introduction

Many computer centers share their resources through some communication linking. For economic reasons most pairs of computer centers are not directly linked so a message or file often travels over several links to get to its destination. For operational reliability most source-destination pairs are connected with two or more distinct message paths.

The choice of which path a message is to follow is a routing decision. This decision might be to use the path with the smallest number of links, provided it is not congested. If the delays resulting from queuing at the computer centers and transmission across the links are significant, then the routing decision might be to use the path with the shortest delay.

These routing decisions require some knowledge of the network. In centralized routing one computer center (with perhaps several backup centers) takes the responsibility of monitoring the network. This center receives the status of the links and the size of the traffic flow from each source to each destination. It sends out to the other centers their routing instructions.

In distributed routing, each center determines the best route based on the status of the adjacent links and on its neighbors' estimates of their distances (number of links, the delay, or whatever) from the destination. More details are given later.

Our basic routing criteria is that the routing should minimize a network cost function. The specific form of this function is given in the next section. The mean delay is a common example of the network cost.

Iterating a routing algorithm should lead to the optimal routing whenever the traffic from each source to each destination is constant. Gallager [77] gave the first distributed routing algorithm that satisfies this criteria. Bertsekas [78] and Gafni [79] generalized Gallager's algorithm and gave different proofs of convergence.

In contrast to centralized routing these distributed algorithms are often guaranteed to converge only if the routing changes are small at each iteration. Presumably, the convergence is also slower than that of centralized algorithms. In chapter three we describe a distributed routing algorithm in which the routing changes are comparable to those of a centralized routing algorithm.

In order to maintain a good routing when the source-destination traffic slowly changes, the routing algorithm is periodically iterated. An algorithm with a good convergence rate should need fewer iterations to give a good routing. If the traffic changes more rapidly, then the algorithm is iterated more often. In the extreme case, the traffic changes so rapidly that the algorithm cannot cope with it.

Empirical tests with a specific network are required to determine how often a routing algorithm should be iterated.

Rudin [80] suggests that these tests should take into account the control used to limit the traffic into the network. Under this flow control a good routing algorithm allows more traffic to enter the network.

The rest of this chapter details our routing problem and gives a simple distributed routing algorithm. The second chapter gives two distributed routing algorithms that make the quadratic approximation of the change in the network cost non-positive. This means that the flow change generated is likely to make the cost function decrease. The third chapter gives a class of routing algorithms in which the network control center is called upon to determine the size of the flow change. The third chapter also shows that this class converges. Appendix B illustrates some sample behavior of the algorithms mentioned in the thesis.

1.1 The Routing Problem

Let N be the set of nodes (switching centers). A duplex communication channel between nodes i and j is interpreted as a pair of directed links (i,j) and (j,i). Let L be the set of directed links.

We will call the traffic destined for node k commodity k and we will let C denote the set of commodities.

The instantaneous description of the network consists of the number of bits of each commondity travelling on each link and the number of bits of each commodity waiting at each node for transmission over a link. This instantaneous description is of little use for routing as during the time it takes a node to describe to its neighbors the state of its queues, the queue lengths usually change significantly.

A less volatile description of the network is the average rate over a given time interval at which each commodity travels over each link and the average rate at which each queue length changes. If the time interval is large enough then, because the queue lengths are finite, the latter rates will be small compared to the former. For example, in ten seconds 500,000 bits might be sent over a link while the queue length at the head of the link changes by 5000 bits. The average rate of change of 500 bits per second in the queue is small compared to average transmission rate of 50,000 bits per second on the link. Consequently, we will treat the rates at which the queue lengths change as zero.

Let F_{ij}^k be the average rate at which commodity k travels over link (i,j). F_{ij}^k is non-negative. Let f_{ij} be the aggregate flow on link (i,j), i.e., $f_{ij} = \Sigma_k F_{ij}^k$. Let c_{ij} be the capacity of link (i,j). Let R_i^k be the average rate at which commodity k originates at node i. R_i^k is non-negative for $i \neq k$. The consumption of commodity k at node k implies

$$R_k^k := -\Sigma_{i \neq k} R_i^k \qquad k \in C \qquad (1.1.1)$$

The conservation of each commodity at each node and our treatment of the queue lengths as zero gives

$$\Sigma_j F_{ij}^k = \Sigma_\ell F_{\ell i}^k + R_i^k \qquad i \in N,\ k \in C \qquad (1.1.2)$$

The network cost will be taken to be the mean transmission cost

$$\frac{1}{r} \Sigma_{ij} f_{ij} t_{ij}(f_{ij}) \qquad (1.1.3)$$

where

$$r := \Sigma_k \Sigma_{i \neq k} R_i^k \qquad (1.1.4)$$

$t_{ij}(f_{ij})$ might be 1, representing the unit use of link (i,j) in which case the network cost is the mean hop length. Alternately, $t_{ij}(f_{ij})$ might be $(c_{ij} - f_{ij})^{-1}$ which is a standard approximate formula for the mean delay on link (i,j) in a store and forward network [Cantor and Gerla 74]. We will write the network cost as

$$J(f) := \Sigma_{ij} J_{ij}(f_{ij}) \qquad (1.1.5)$$

The link cost $J_{ij}(f_{ij})$ is a function of the flow on that link. Our routing problem formulation is

$$\begin{aligned} \text{minimize } & \Sigma_{ij} J_{ij}(f_{ij}) \\ \text{subject to } & f_{ij} = \Sigma_k F_{ij}^k \\ & \Sigma_m F_{im}^k - \Sigma_n F_{ni}^k = R_i^k \\ & f_{ij} \leq c_{ij} \\ & F_{ij} \geq 0 \\ & (i,j) \in \mathcal{L},\ k \in \mathcal{C} \end{aligned} \qquad (1.1.6)$$

The set of feasible flows is defined to be the set of variables f,F that satisfies the constraints of (1.1.6). This thesis assumes that on the set of feasible flows the network cost is twice continuously differentiable with positive partial derivatives and non-negative second partial derivatives.

Almost all of the results about the routing alogrithms in chapters two and three hold only when the input flow R is constant. Rather than preface many remarks with the clause, "Assuming R is constant...," we here assume that R is constant in the rest of the thesis.

As stated in appendix A, the optimal flow of (1.1.6) is the one in which the flow travels along the path of the least incremental cost. If the derivative of $J_{ij}(f_{ij})$ is taken to be the distance of the link (i,j) then the optimal flow is a shortest distance flow. We denote this link distance as

$$g_{ij} := \frac{\partial J_{ij}(f_{ij})}{\partial f_{ij}} \qquad (i,j)\in L \qquad (1.1.7)$$

By our assumptions about the network cost function we have

$$g_{ij} > 0 \qquad (i,j)\in L \qquad (1.1.8)$$

1.2 <u>A Distributed Routing Algorithm</u>

We introduce the basic details of distributed routing with the following routing algorithm. Recall that commodity k is the flow destined for node k. As we often describe what is happening one commodity at a time, we will just as often find it convenient to omit the superscript k. If $(i,j)\in L$ or $(j,i)\in L$ then we say that j is a neighbor of i.

In the following algorithm every node has a favored neighbor for each commodity. Initially, we let this be the neighbor on any path with the fewest links to the destination. For each commodity, V_i is the distance from i to the destination via the favored neighbors. Each node i selects as its new favored neighbor the neighbor m that minimizes $g_{im}+V_m$. (g_{im} is given in (1.1.7)). (We will later show that the new favored neighbors determine a tree directed into the destination.) The proposed flow F' is the one that travels to the destination via the new favored neighbors. The new flow generated is that convex combination of the present flow and the proposed flow that minimizes the network cost.

The following describes one iteration of the algorithm. We assume that the initial flow is feasible.

I. The following steps are done for each commodity.

A. The destination sends the signal "$V_{dest} = 0$" to its neighbors.

B. Each node i waits until it receives V_n from its favored neighbor n. Then node i sends the following to its neighbors.

$$V_i = g_{in} + V_n$$

C. Each node i waits until it receives V_j from every neighbor j. Then it waits until it receives F'_{ji}

from every neighbor j for which $V_j > V_i$. Node i automatically assumes $F'_{ji} = 0$ for those neighbors j for which $V_j \leq V_i$. Node i then determines its new favored neighbor m by

$$m = \arg\text{-}\min_j \{g_{ij} + V_j\}$$

The proposed flow is $F'_{im} = \Sigma_\ell F'_{\ell i} + R_i$ and $F'_{ij} = 0$ if $j \neq m$. F'_{ij} is sent to node j if $V_i > V_j$.

II. Each node i sends the set $\{f'_{ij}(=\Sigma_k F'^k_{ij})\}$ to every node.

III. Every node determines the γ, $0 \leq \gamma \leq 1$, that minimizes $J((1-\gamma)f+\gamma f')$ while satisfying $(1-\gamma)f+\gamma f' \leq c$. For each commodity and link (i,j) the new flow is $(1-\gamma)F_{ij} + \gamma F'_{ij}$.

In that last step a central node could receive f', determine γ, and send this information out to all nodes. This would reduce the communication overhead, but would also introduce a delay before every node learns the value of γ.

We now briefly check that the above algorithm generates a feasible flow. For each commodity and each node i we have for the n and m of step B and C,

$$\begin{aligned} V_i &= g_{in} + V_n \\ &\geq g_{im} + V_m \\ &> V_m \end{aligned} \tag{1.2.1}$$

where the last inequality comes from (1.1.8). Thus, the new favored neighbors determine a tree rooted into the destination. The proposed flow travels on this tree and satisfies all of the constraints of (1.1.6) except, possibly, $f' \leq c$. Then, since f is feasible, any flow satisfying $(1-\gamma)f + \gamma f' \leq c$ with $0 \leq \gamma \leq 1$ is feasible. This shows that the algorithm is feasible.

When the link costs depend on the flow and the network is moderately to heavily loaded, the optimal flow usually has branches, i.e., multiple paths. In this situation a routing algorithm should not only indicate which path the flow is to be shifted to but also how much of the flow is to be shifted. The routing algorithms of chapter two do this job.

We end this chapter with an example. We start with the network of figure 1.2.1 and use only one commodity, that destined for node d. The link capacities are $c_{ij} = 5$. The input flows are $R_1 = 3$, $R_2 = 2$, (and $R_d = -5$). The sum of the input flows is $r = 5$. The link cost functions are

$$J_{ij}(f_{ij}) = \frac{1}{5} \cdot \frac{f_{ij}}{5-f_{ij}} \tag{1.2.2}$$

For the initial flow we have $F_{1d} = 3$, $F_{2d} = 2$. With the following

$$\frac{\partial J_{ij}(f_{ij})}{\partial f_{ij}} = \frac{1}{(5-f_{ij})^2} , \tag{1.2.3}$$

we have for the initial flow, $g_{1d} = 1/4$, $g_{2d} = 1/9$, $g_{12} = g_{21} = 1/25$. Applying step IB of the algorithm we have $V_1 = 1/4$,

$V_2 = 1/9$. Node d remains the favored node of 2. Node 2 becomes the favored node of 1 because $1/4 > 1/25 + 1/9$. The proposed flow is $F'_{12} = 3$, $F'_{2d} = 5$. In step III, the optimal γ will be found to be .3232. The new flow turns out to be the optimal flow with cost .4178. The initial flow had .4333 for its network cost. If each node had minimized its delay from the destination then the scaling γ above would have been .4398 and the resulting network cost is .4198.

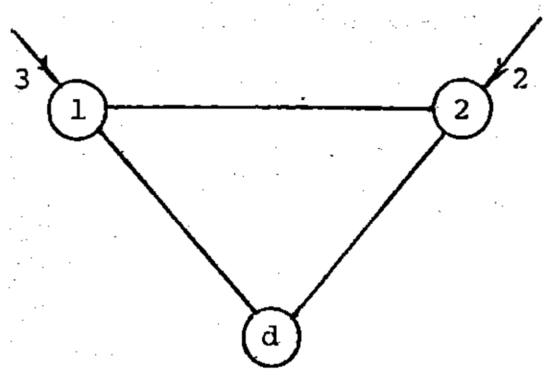

Figure 1.2.1

Note 1.N.1. Schwartz and Stern [80] summarizes the routing schemes used in present day networks. None except ARPANET uses distributed routing. Many do have distributed communication for notification of congestion. In response to congestion, these networks change the routing to some predetermined path. None of the networks optimizes the network delay. The network that comes closest to this is the ARPANET [McQuillan et alia, 80]. It takes the present delays in the network, finds the paths with the smallest delay (not incremental delay) and uses these paths ignoring the fact that the new routing would change the link delays.

Note 1.N.2. For centralized routing there are many techniques to find the optimal routing. Among them are Cantor and Gerla [74], Frank and Chou [71], the flow deviation method [Fratta, Gerla, and Kleinrock 73] and Schwartz and Cheung [76]. The routing algorithm given in section 1.2 is a distributed approximation of the flow deviation method. In the exact flow deviation method the proposed flow F' is sent down the path with the shortest distance wrt g.

Note 1.N.3. Agnew [76] indicates that the difference between using the paths with the shortest delay and using the paths with the shortest incremental delay is largest at medium loadings. Rudin [80] indicates that the effect of using frequent routing determinations is most beneficial at medium loadings. The reason is that at low loadings the flow usually follow the path with the fewest links. At high loadings, the highest cost links form a cut set and the routing either avoids the cut set or balances the flow through the cut set so that the cost of the links are about the same.

Note 1.N.4. Our routing criteria is that the routing should become optimal if the source-destination traffic remains constant. Since this traffic changes one might try to improve on the routing in between routing determinations. Rudin [76] reports some benefit in using something close to the actual delay to the neighbor plus the reported delay from the neighbor to the destination in determining the shortest path. One might also predict the interim delay as a function of the flow sent to the destination by means the second derivative methods of chapter two in this thesis.

Chapter II.

Second Order Routing

The routing algorithms of this chapter generate a flow change $\tilde{F}$ that reduces the quadratic approximation of the cost change. The convergence of these algorithms is covered in the next chapter.

2.1 Notation

We will be using the vector version of formulation (1.1.6) given in the first chapter along with some new quantities. Let N be the number of nodes, L be the number of links, and C be the number of commodities. From section 1.1 we have C=N. The commodity flow F^k is $L\times 1$ and the input flow R^k is $N\times 1$. For the most part we work with one commodity so it will be convenient to let F and R be the flow of a fixed commodity. The destination of this commodity will be called node dest.

Let the following $N\times L$ node-link incidence matrices be defined.

$$E^+_{n,ij} := \begin{cases} 1 & \text{if} \quad n=i \\ 0 & \text{if} \quad n\neq i \end{cases}$$

$$E^-_{n,ij} := \begin{cases} 1 & \text{if} \quad n=j \\ 0 & \text{if} \quad n\neq j \end{cases} \qquad n\in\mathcal{N},\ (i,j)\in\mathcal{L} \qquad (2.1.1)$$

$$E_{n,ij} := E^+_{n,ij} - E^-_{n,ij}$$

The conservation equation (1.1.2) is then $E^+F = E^-F+R$. Let the node flow T be defined by

$$T := E^{+}F = E^{-}F + R \qquad (2.1.2)$$

T is N×1. A routing variable ψ is defined to be any non-negative L×N matrix that satisfies

$$\Sigma_j \psi_{ij,n} = \left\{ \begin{array}{ll} 1 & \text{if } n=i\neq \text{dest} \\ 0 & \text{if } n\neq i,\ i=\text{dest, or } n=\text{dest} \end{array} \right\} (i,j)\in L,\ n\in N \qquad (2.1.3)$$

Let the routing fraction ϕ be any (usually unique) routing variable such that if $T_i \neq 0$ then $\phi_{ij,i} = F_{ij}/T_i$.

If ϕ_{ij} is taken to be the row of ϕ that corresponds to link (i,j), then from the definition of a routing variable, ϕ_{ij} is non-zero only at $\phi_{ij,i}$. We have $F_{ij} = \phi_{ij}T = \phi_{ij,i}\ T_i$. It will be convenient to take ϕ_{ij} to be the element $\phi_{ij,i}$. Then $F_{ij} = \phi_{ij}T_i$. We will take ϕ_i to be the column of ϕ that corresponds to node i. The non-zero elements of $\phi_i T_i$ are the flows that leave node i.

$$F = \phi T \qquad (2.1.4)$$

Let $\tilde{F}$ be a change in F. Let $F' = F + \tilde{F}$ be the new flow after making the change. Let T' and ϕ' correspond to F'. Let $\tilde{T} = T' - T$ and $\tilde{\phi} = \phi' - \phi$. From (2.1.2) and (2.1.4) we have the basic equations involving these quantities. As discussed near the end of section 1.1, R is assumed to be constant.

$$T' = E^{+}F' = E^{-}F' + R \qquad (2.1.5)$$

$$\tilde{T} = E^{+}\tilde{F} = E^{-}\tilde{F} \qquad (2.1.6)$$

$$F' = \phi' T' \qquad (2.1.7)$$

$$\tilde{F} = \phi' T' - \phi T \tag{2.1.8}$$

$$= \tilde{\phi} T + \tilde{\phi}\tilde{T} + \phi\tilde{T} \tag{2.1.9}$$

The last equation comes from the expansion of ϕ' and T' in the previous equation.

The definition of ψ implies that a change in any routing variable such as ϕ satisfies

$$\Sigma_j \tilde{\phi}_{ij} = 0 \qquad \phi_{ik} + \tilde{\phi}_{ik} \geq 0 \qquad (i,k) \in L \tag{2.1.10}$$

If $\phi_{ij} > 0$ we say that j is a <u>downstream neighbor</u> of i. A node k is <u>downstream</u> of i if it is a downstream neighbor of i or if it is downstream of a downstream neighbor of i. ϕ is said to be <u>loopfree</u> if no node is downstream of itself.

Let $Q(\tilde{f})$ be the quadratic appoximation of the cost difference $J(f+\tilde{f}) - J(f)$.

$$Q(\tilde{f}) := \Sigma_{ij} g_{ij} \tilde{f}_{ij} + \frac{1}{2} \Sigma_{ij} h_{ij} \tilde{f}_{ij}^2 \tag{2.1.11}$$

where

$$g_{ij} := \frac{\partial J_{ij}(f_{ij})}{\partial f_{ij}} \tag{2.1.12}$$

$$h_{ij} := \frac{\partial^2 J_{ij}(f_{ij})}{\partial f_{ij}^2} \tag{2.1.13}$$

By our assumption about the network cost, stated in (1.1.6), $g_{ij} > 0$ and $h_{ij} \geq 0$.

Letting g and h be $1 \times L$ vectors and the square of a vector be the vector of squares, (2.1.11) may be rewritten as

$$Q(\tilde{f}) := g\tilde{f} + \frac{1}{2} h\tilde{f}^2 \qquad (2.1.14)$$

Because $Q(\tilde{f})$ is quadratic it is symmetric about its minimum. Since $Q(0) = 0$, if $\tilde{f}$ minimizes $Q(\tilde{f})$ then $Q(2\tilde{f})=0$. We are not able to minimize $Q(\tilde{f})$ directly with a distributed routing algorithm. When we upper bounded $Q(\tilde{f})$ by other quadratics and minimized those bounds the only solid thing we could say about the result was that $Q(\tilde{f}) \leq 0$. In that case, we rather have $Q(2\tilde{f}) \leq 0$. The two distributed routing algorithms of this chapter generate a flow change $\tilde{f}$ such that $Q(2\tilde{f}) \leq 0$.

We have

$$\begin{aligned} Q(2\tilde{f}) &= 2g\tilde{f} + 2\Sigma_{ij}h_{ij}\tilde{f}_{ij}^2 \\ &= 2g\Sigma_k\tilde{F}^k + 2\Sigma_{ij}h_{ij}(\Sigma_k\tilde{F}_{ij}^k)^2 \\ &\leq 2g\Sigma_k\tilde{F}^k + 2\Sigma_{ij}h_{ij}C\Sigma_k(\tilde{F}_{ij}^k)^2 \\ &= 2\Sigma_k(g\tilde{F}^k + Ch(\tilde{F}^k)^2) \qquad (2.1.15) \end{aligned}$$

The routing algorithms of this chapter make $Q(2\tilde{f})$ non-positive by making $g\tilde{F} + Ch\tilde{F}^2$ non-positive for every commodity.

2.2 An Algorithm Using Second Derivatives

As we shall show subsequently, the following algorithm has the property that $g\tilde{F} + Ch\tilde{F}^2$ is non-positive. For the first iteration of this algorithm we assume that ϕ is loopfree. For each iteration the following steps are done for each commodity.

IA. The destination sends the signal "$G_{dest}=0$, $H_{dest}=0$, $S_{dest}=0$" to its neighbors.

IB. When each node i receives G_j and H_j and S_j from every j such that $\phi_{ij} > 0$, node i passes the following values to its neighbors

$$G_i = \Sigma_j (g_{ij}+G_j)\phi_{ij}$$

$$H_i = \Sigma_j (h_{ij}\phi_{ij}+H_j)\phi_{ij}$$

$$S_i = \max\{G_i, \max_{j:\phi_{ij}>0} S_j\}$$

IC. When each node i receives the G, H, and S values from every neighbor and the size of the new incoming flow F'_{ki} from every neighbor k such that $S_k > S_i$ or $\phi_{ki} > 0$ then node i determines the new node flow $T'_i=\Sigma_k F'_{ki}$ and the $\tilde{\phi}$ which minimizes the following

$$\Sigma_j (g_{ij}+G_j)\tilde{\phi}_{ij} + \frac{1}{2} CL\Sigma_j (h_{ij}+H_j)\tilde{\phi}^2_{ij}T'_i \qquad (2.2.1)$$

such that $\Sigma_j\tilde{\phi}_{kj}=0$ and $\phi_{ij}+\tilde{\phi}_{ij} \geq 0$ and $\tilde{\phi}_{ij}=0$ if $S_i \leq S_j$ and $\phi_{ij}=0$. The new flow is $F'_{ij}=(\phi_{ij}+\tilde{\phi}_{ij})T'_i$ This is passed down to each node j.

The <u>node distance</u> G_i is the average distance over which the input flow R_i travels to reach the destination. Note 2.N.2 shows that G_i is the marginal change in J with respect to R_i when ϕ is fixed. The <u>watershed distance</u> S_i is the largest node distance at node i or downstream of node i. Its only role is to prevent a deadlock from occurring in the algorithm.

ϕ is assumed to be loopfree so there is no deadlock in step IB. To make sure that there is no deadlock in IB in the next iteration we must make sure that $\phi' = \phi + \tilde{\phi}$ is loopfree.

If either (i) $\phi_{ij} > 0$ or (ii) $(i,j) \in L$ and $S_i > S_j$ then we say that j is a downhill neighbor of i or that i is an uphill neighbor of j. In step IC node i receives the size F'_{ki} from every uphill neighbor and constrains $\tilde{\phi}_{ij}=0$, i.e. $\phi_{ij}+\tilde{\phi}_{ij} = 0$, if j is not a downhill neighbor. We say that k is downhill of i if it is a downhill neighbor of i or is downhill of a downhill neighbor of i. No deadlock occurs in IC if no node is downhill of itself. Also ϕ'_{ij} is non-zero only if j is a downhill neighbor of i so if no node is downhill of itself then ϕ' is loopfree. Thus, to show that the algorithm is feasible we only have to show that no node is downhill of itself.

By the definition of downhill and the definition of S (in IB), if j is a downhill neighbor of i then $S_j \leq S_i$ with equality only if j is a downstream neighbor of i. Thus, if j is downhill of i then $S_j \leq S_i$ with equality only if j is downstream of i. Thus, a node is downhill of itself only if it is downstream of itself. But ϕ is loopfree so no node is downhill of itself. This proves that the algorithm is feasible.

In step IC, node i needs to know T'_i before it can determine $\tilde{\phi}$. Rather than have node i take the time necessary to measure T'_i we have every uphill neighbor k send the size F'_{ki}

to node i. Calculating the new flow is also useful if several iterations of the algorithm per measurement interval is desired.

We mention here the effect T_i' has on the optimal $\tilde{\phi}_i$. If T_i' is large then the squared term in (2.2.1) is penalized so the optimal $\tilde{\phi}_i$ is small. If T_i' is small then the optimal $\tilde{\phi}_i$ will be large.

Remark 2.2.1. Algorithm I makes $g\tilde{F} + Ch\tilde{F}^2$ non-positive.

Proof. Algorithm I minimizes (2.2.1) which is given in step IC. Thus, any change from the optimal $\tilde{\phi}$ to, say, $\tilde{\psi}$ where $\tilde{\psi}$ satisfies the constraints of IC, is a non-descent change in (2.2.1).

$$\Sigma_i [g_{ij}+G_j+CL(h_{ij}+H_j)\tilde{\phi}_{ij}T_i'](\tilde{\psi}_{ij}-\tilde{\phi}_{ij}) \geq 0 \qquad (2.2.2)$$

The expression in the brackets is the gradient of (2.2.1). At $\tilde{\psi} = 0$, the above inequality reduces to

$$\Sigma_j (g_{ij}+G_j)\tilde{\phi}_{ij} + CL\Sigma_j (h_{ij}+H_j)\tilde{\phi}_{ij}^2 T_i' \leq 0 \qquad (2.2.3)$$

Multiplying by T_i' and summing over i gives

$$(g+GE^-)\tilde{\phi}T' + CL(h+HE^-)(\tilde{\phi}T')^2 \leq 0 \qquad (2.2.4)$$

We now develop some expressions for G and H. From IB,

$$G_i = \Sigma_j g_{ij}\phi_{ij} + \Sigma_j G_j \phi_{ij} \qquad (2.2.5)$$

In vector form, this is

$$G = g\phi + GE^-\phi \qquad (2.2.6)$$

Subtracting the last term, gives

$$G(I-E^{-}\phi) = g\phi \tag{2.2.7}$$

Let the following be defined.

$$\theta_N := (I-E^{-}\phi)^{-1} \tag{2.2.8}$$

Note 2.N.1 shows that the inverse exists and that its terms are non-negative and less than one. θ_N is N×N. $\theta_{k,i}$ can be interpreted as the fraction of R_i that appears at node k. Multiplying (2.2.7) by θ_N gives

$$G = g\phi\theta_N \tag{2.2.9}$$

That is,

$$G_n = \Sigma_{ij} g_{ij} \phi_{ij} \theta_{i,n} \tag{2.2.10}$$

Similarly, for H

$$H_n = \Sigma_{ij} h_{ij} \phi_{ij}^2 \theta_{i,n} \tag{2.2.11}$$

With (2.2.9), the first term of (2.2.4) is

$$\begin{aligned}(g+GE^{-})\tilde{\phi}T' &= (g+g\phi\theta_N E^{-})\tilde{\phi}T' \\ &= g(I+\phi\theta_N E^{-})\tilde{\phi}T' \end{aligned} \tag{2.2.12}$$

We now show that this is $g\tilde{F}$. From (2.1.9)

$$\begin{aligned}\tilde{F} &= \tilde{\phi}T + \tilde{\phi}\tilde{T} + \phi\tilde{T} \\ &= \tilde{\phi}T' + \phi\tilde{T}\end{aligned} \tag{2.2.13}$$

Putting this in $\tilde{T} = E^{-}\tilde{F}$ gives

$$\tilde{T} = E^{-}\tilde{\phi}T' + E^{-}\phi\tilde{T}$$

Subtracting the last term from both sides and then multiplying by θ_N gives

$$\tilde{T} = \theta_N E^{-}\tilde{\phi}T' \qquad (2.2.14)$$

Putting this in (2.2.13) gives

$$\tilde{F} = \tilde{\phi}T' + \phi\theta_N E^{-}\tilde{\phi}T' \qquad (2.2.15)$$

$$= (I+\phi\theta_N E^{-})\tilde{\phi}T' \qquad (2.2.16)$$

Using this in (2.2.12) gives

$$(g+GE^{-})\tilde{\phi}T' = g\tilde{F} \qquad (2.2.17)$$

Thus, (2.2.4) becomes

$$g\tilde{F} + CL(h+HE^{-})(\tilde{\phi}T')^2 \leq 0 \qquad (2.2.18)$$

We now show that $Ch\tilde{F}^2$ is less than the second term above. Using (2.2.15),

$$Ch\tilde{F}^2 = Ch[\tilde{\phi}T'+\phi\theta_N E^{-}\tilde{\phi}T']^2$$

$$= C\Sigma_{ij}h_{ij}[\tilde{\phi}_{ij}T'_i+\Sigma_{mn}\phi_{ij}\theta_{i,n}\tilde{\phi}_{mn}T'_m]^2 \qquad (2.2.19)$$

Since $T'_{dest}=0$, there are no more than L term in the brackets. Upper bounding the square of sums with L times the sum of squares gives

$$Ch\tilde{F}^2 \le CL\Sigma_{ij}h_{ij}[(\tilde{\phi}_{ij}T_i')^2 + \Sigma_{mn}\phi_{ij}^2\theta_{i,n}^2(\tilde{\phi}_{mn}T_m')^2]$$

$$= CL\Sigma_{ij}h_{ij}(\tilde{\phi}_{ij}T_i')^2 + CL\Sigma_{ij}h_{ij}\Sigma_{mn}\phi_{ij}^2\theta_{i,n}^2(\tilde{\phi}_{mn}T_m')^2$$

$$= CLh(\tilde{\phi}T')^2 + CL\Sigma_{mn}\Sigma_{ij}h_{ij}\phi_{ij}^2\theta_{i,n}^2(\tilde{\phi}_{mn}T_m')^2 \qquad (2.2.20)$$

From the fact $0 \le \theta_{i,n} \le 1$ we have $\theta_{i,n} \ge \theta_{i,n}^2$. Using this and the fact that h_{ij} is non-negative in (2.2.11) gives

$$H_n \ge \Sigma_{ij}h_{ij}\phi_{ij}^2\theta_{i,n}^2 \qquad (2.2.21)$$

Using this in (2.2.20) gives

$$Ch\tilde{F}^2 \le CLh(\tilde{\phi}T')^2 + CL\Sigma_{mn}H_n(\tilde{\phi}_{mn}T_m')^2$$

$$= CLh(\tilde{\phi}T')^2 + CLHE^-(\tilde{\phi}T')^2$$

$$= CL(h+HE^-)(\tilde{\phi}T')^2 \qquad (2.2.22)$$

Using this in (2.2.18) then gives the remark.

The next remark says that there exists a flow cost below which alogrithm I makes the network cost decrease monotonely.

Remark 2.2.2. Suppose that the second derivative of each J_{ij} is positive and that the initial flow f^0 is such that, for every link (i,j),

$$\max_{f}\left\{\frac{\partial^2 J_{ij}(f_{ij})}{\partial f_{ij}^2}\middle| J(f) \leq J(f^0)\right\} \leq$$

$$2\min_{f}\left\{\frac{\partial^2 J_{ij}(f_{ij})}{\partial f_{ij}^2}\middle| J(f) \leq J(f^0)\right\} \quad (2.2.23)$$

Then algorithm I makes the network cost decrease monotonely.

Proof. Let f be any flow such that $J(f) \leq J(f^0)$ and, for f, suppose that algorithm I generates $\tilde{f}$. Suppose to the contrary that $J(f+\tilde{f}) > J(f)$. Because of the strict inequality $J(f+\tilde{f}) > J(f)$, we have $\tilde{f}\neq 0$. With remark 2.2.1 and equations (2.1.15) and (2.1.14), algorithm I is seen to give

$$g\tilde{f} + h\tilde{f}^2 \leq 0 \quad (2.2.24)$$

By the first assumption of the remark we have $h_{ij} > 0$ for all links (i,j). Since we also have $\tilde{f}\neq 0$, gf is negative.

$$g\tilde{f} \leq -h\tilde{f}^2 < 0$$

This means $\tilde{f}$ is in a descent direction. Since J is continuous, there exists an α, $0 < \alpha < 1$, such that $J(f+\alpha\hat{f}) = J(f)$. A Taylor series expansion of $J(f+\alpha\tilde{f})$ gives

$$J(f+\alpha\tilde{f}) = J(f) + \alpha g\tilde{f} + \frac{\alpha^2}{2}\Sigma_{ij}\frac{\partial^2 J_{ij}(\xi_{ij})}{\partial f_{ij}^2}\tilde{f}_{ij}^2 \quad (2.2.25)$$

where ξ_{ij} is between f_{ij} and $f_{ij}+\alpha\tilde{f}_{ij}$. We have $J(f+\xi) \leq J(f^0)$. Then (2.2.23) allows us to say

$$\frac{\partial^2 J_{ij}(\xi_{ij})}{\partial f_{ij}^2} \leq 2\ \frac{\partial^2 J_{ij}(f_{ij})}{\partial f_{ij}^2}$$

$$= 2h_{ij} \qquad (2.2.26)$$

Using this inequality in (2.2.25) leads to

$$0 = J(f+\alpha\tilde{f}) - J(f) \leq \alpha g\tilde{f} + \alpha^2 h\tilde{f}^2 \qquad (2.2.27)$$

$$< \alpha(g\tilde{f}+h\tilde{f}^2)$$

$$\leq 0 \qquad (2.2.28)$$

This is 0 < 0, a contradiction. Therefore, the network cost decreases monotonely.

2.3 An Algorithm Not Using Second Derivatives

Let P be a number greater than max h_{ij}. The following algorithm is less precise than the previous one as it uses P instead of the whole vector h. It does have two advantages. H is not passed through the network. Also, the optimizing $\tilde{\phi}_i$ is independent of $\tilde{T}_i$, avoiding the need to pass the size of the flow change through the network. Node i simply calculates $\phi_i' = \phi_i + \tilde{\phi}_i$ and proportions out with ϕ' the traffic that comes into the node.

In the first iteration of the algorithm we assume ϕ is loopfree.

IIA. The destination sends the signal "$G_{dest}=0$, $S_{dest}=0$" to its neighbors.

IIB. When each node i receives G_j and S_j from every j such that $\phi_{ij} > 0$, then node i passes the following values to its neighbors

$$G_i = \Sigma_j (g_{ij}+G_j)\phi_{ij}$$

$$S_i = \max\{G_i, \max_{j:\phi_{ij}>0} S_j\}$$

IIC. When each node i receives the G and S values from every neighbor it determines the $\tilde{\phi}_i$ which minimizes the following

$$\Sigma_j (g_{ij}+G_j)\tilde{\phi}_{ij} + \frac{1}{4} \text{PCLN}\Sigma_j \tilde{\phi}^2_{ij} T_i \qquad (2.3.1)$$

such that $\Sigma_j \tilde{\phi}_{ij}=0$, $\phi_{ij}+\tilde{\phi}_{ij} \geq 0$, and $\tilde{\phi}_{ij}=0$ if $S_i \leq S_j$ and $\phi_{ij}=0$.

The new flow is $F'_{ij} = (\phi_{ij}+\tilde{\phi}_{ij})T'_i$ where $T'_i = \Sigma_k F'_k$. The demonstration that this algorithm is feasible is the same as for the previous algorithm except that we do not have to worry about a deadlock in step IIC.

Remark 2.3.1. Algorithm II makes $g\tilde{F} + PC|\tilde{F}|^2$ non-positive. Note that a non-positive $g\tilde{F}+PC|\tilde{F}|^2$ makes $g\tilde{F}+Ch\tilde{F}^2$ non-positive.
Proof. Using the same argument as for (2.2.3), the optimal $\tilde{\phi}_i$ satisfies

$$\Sigma_j (g_{ij}+G_j)\tilde{\phi}_{ij} + \frac{1}{2}PCLN\Sigma_j \tilde{\phi}^2_{ij}T_i \leq 0 \qquad (2.3.2)$$

Multiplying by T'_i and then summing over i gives.

$$(g+GE^-)\tilde{\phi}T' + \frac{1}{2}PCLN\Sigma_i |\tilde{\phi}_i|^2 T_i T'_i \leq 0 \qquad (2.3.3)$$

By (2.2.17) the first term is $g\tilde{F}$.

$$g\tilde{F} + \frac{1}{2} PCLN\Sigma_i |\tilde{\phi}_i|^2 T_i T'_i \leq 0 \qquad (2.3.4)$$

For any variable x, let $x^+ = \max\{x,0\}$ and $x^- = \max\{0,-x\}$. We have $x = x^+ - x^-$.

We have $T_i \geq T_i - \tilde{T}_i^-$ and $T'_i = T_i + \tilde{T}_i \geq T_i - \tilde{T}_i^-$. Using both of these inequalities in (2.3.4) gives

$$g\tilde{F} + \frac{1}{2} PCLN|\tilde{\phi}(T-\tilde{T}^-)|^2 \leq 0 \qquad (2.3.5)$$

We now show that $|\tilde{F}|^2 \leq \frac{1}{2} LN|\tilde{\phi}(T-\tilde{T}^-)|^2$ and this in the above will prove the remark. From (2.1.9),

$$\tilde{F} = \tilde{\phi}T + \tilde{\phi}\tilde{T} + \phi\tilde{T}$$

Expanding $\tilde{T}$ into $\tilde{T}^+ - \tilde{T}^-$ gives

$$\tilde{F} = \tilde{\phi}T - \tilde{\phi}\tilde{T}^- + \tilde{\phi}\tilde{T}^+ + \phi\tilde{T}^+ - \phi\tilde{T}^-$$

$$= \tilde{\phi}(T-\tilde{T}^-) + \phi'\tilde{T}^+ - \phi\tilde{T}^- \qquad (2.3.6)$$

From this we get,

$$\tilde{F}^- \leq \tilde{\phi}^-(T-\tilde{T}^-) + \phi\tilde{T}^- \qquad (2.3.7)$$

With $\tilde{T}^- \leq E^-\tilde{F}^-$,

$$\tilde{F}^- \leq \tilde{\phi}^-(T-\tilde{T}^-) + \phi E^-\tilde{F}^-$$

Subtracting the last term from both sides,

$$(I-\phi E^{-})\tilde{F}^{-} \leq \tilde{\phi}^{-}(T-\tilde{T}^{-}) \tag{2.3.8}$$

Let the following be defined.

$$\theta_L := (I-\phi E^{-})^{-1} \tag{2.3.9}$$

Note 2.N.1 shows that the matrix inverse exists and that its terms are non-negative and no greater than one. θ_L is L×L. $\theta_{ij,mn}$ is the fraction of $\phi_{mn}R_m$ that appears on link (i,j).

Multiplying (2.3.8) by θ_L gives

$$\tilde{F}^{-} \leq \theta_L\tilde{\phi}^{-}(T-\tilde{T}^{-}) \tag{2.3.10}$$

Elementwise, this is

$$\tilde{F}^{-}_{ij} \leq \Sigma_{mn}\theta_{ij,mn}\tilde{\phi}^{-}_{mn}(T_m-\tilde{T}^{-}_m) \tag{2.3.11}$$

Squaring both sides and summing over ij gives

$$|\tilde{F}^{-}|^2 \leq \Sigma_{ij}(\Sigma_{mn}\theta_{ij,mn}\tilde{\phi}^{-}_{mn}(T_m-\tilde{T}^{-}_m))^2 \tag{2.3.12}$$

Using Minkowski's inequality, $\Sigma_j(\Sigma_k x_{jk}^{1/2})^2 \leq (\Sigma_k(\Sigma_j x_{jk})^{1/2})^2$ in the form $\Sigma_j(\Sigma_k x_{jk})^2 \leq (\Sigma_k(\Sigma_j x_{jk}^2)^{1/2})^2$ gives

$$|\tilde{F}^{-}|^2 \leq (\Sigma_{mn}(\Sigma_{ij}\theta^2_{ij,mn}(\tilde{\phi}^{-}_{mn}(T_m-\tilde{T}^{-}_m))^2)^{1/2})^2 \tag{2.3.13}$$

Since $0 \leq \theta_{ij,mn} \leq 1$ we have

$$\Sigma_{ij}\theta^2_{ij,mn} \leq \Sigma_{ij}\theta_{ij,mn}$$

With equation (2.N1.12) found in note 2.N.1 the above becomes

$$\Sigma_{ij}\theta^2_{ij,mn} \leq N \tag{2.3.14}$$

Using this in (2.3.13) gives

$$|\tilde{F}^-|^2 \leq N(\Sigma_{mn}\tilde{\phi}^-_{mn}(T_m-\tilde{T}^-_m))^2 \tag{2.3.15}$$

With $\Sigma_n\tilde{\phi}^-_{mn} = \frac{1}{2}\Sigma_n|\tilde{\phi}_{mn}|$, the above becomes

$$|\tilde{F}^-|^2 \leq \frac{N}{4}(\Sigma_{mn}|\tilde{\phi}_{mn}|(T_m-\tilde{T}^-_m))^2 \tag{2.3.16}$$

We do the same development for $\tilde{F}^+$. From (2.3.6),

$$\tilde{F}^+ \leq \tilde{\phi}^+(T-\tilde{T}^-) + \phi'\tilde{T}^+ \tag{2.3.17}$$

With $\tilde{T}^+ \leq E^-\tilde{F}^+$,

$$\tilde{F}^+ \leq \tilde{\phi}^+(T-T^-) + \phi'E^-\tilde{F}^+ \tag{2.3.18}$$

Subtracting the last term from both sides,

$$(I-\phi'E^-)\tilde{F}^+ \leq \tilde{\phi}^+(T-\tilde{T}^-) \tag{2.3.19}$$

Now we define the following

$$\theta'_L := (I-\phi'E^-)^{-1} \tag{2.3.20}$$

It has properties similar to θ_L. The next few steps parallel (2.3.9) to (2.3.15). This gives

$$|\tilde{F}^+|^2 \leq N(\Sigma_{mn}\tilde{\phi}^+_{mn}(T_m-\tilde{T}^-_m))^2 \tag{2.3.21}$$

Using $\Sigma_n\tilde{\phi}^+_{mn} = \frac{1}{2}\Sigma_n|\tilde{\phi}_{mn}|$,

$$|\tilde{F}^+|^2 \leq \frac{N}{2}(\Sigma_{mn}|\tilde{\phi}_{mn}|(T_m-\tilde{T}^-_m))^2 \tag{2.3.22}$$

Using this and (2.3.16) in $|\tilde{F}|^2 = |\tilde{F}^+|^2 + |\tilde{F}^-|^2$ gives

$$|\tilde{F}|^2 \leq \frac{N}{2} \left(\Sigma_{mn} |\tilde{\phi}_{mn}| (T_m - \tilde{T}_m)\right)^2 \tag{2.3.23}$$

$$\leq \frac{NL}{2} \Sigma_{mn} (\tilde{\phi}_{mn} (T_m - \tilde{T}_m))^2$$

$$\leq \frac{NL}{2} |\tilde{\phi}(T - \tilde{T}^-)|^2 \tag{2.3.24}$$

This in (2.3.5) gives the remark.

Note 2.N.1. The routing fraction ϕ is loopfree if there does not exist a loop $(n_1, n_2, \ldots, n_m, n_{m+1} = n_1)$ on which $\phi_{n_i n_{i+1}} > 0$ for $i = 1,2,\ldots,m$. This note shows that if the routing fraction is loopfree then for a given R there is a unique F and T.

From EF=R and $E=E^+-E^-$ and then from $F=\phi T$ and $T=E^+F$ comes

$$R = E^+F - E^-F$$

$$= T - E^-\phi T$$

$$= (I-E^-\phi)T \qquad (2.N1.1)$$

If $I-E^-\phi$ is invertible then T will be unique. We show the invertibility. We have $E_j^-\phi_i = \phi_{ij}$

$$(E^-\phi)^2_{j,i} = \Sigma_k \phi_{ik}\phi_{kj} \qquad (2.N1.2)$$

Inspection of this leads to the following interpretation. $(E^-\phi)^n_{j,i}$ is the fraction of R_i that appears at node j after travelling on exactly n links. Since ϕ is loopfree, $(E^-\phi)^N=0$. We have

$$(I-E^-\phi)(I+E^-\phi+(E^-\phi)^2 + \ldots + (E^-\phi)^{N-1}) = I - (E^-\phi)^N = I$$

Since the product on the left hand side equals the identity matrix, we get

$$(I-E^-\phi)^{-1} = I + E^-\phi + (E^-\phi)^2 + \ldots + (E^-\phi)^{N-1} \qquad (2.N1.3)$$

This shows that $(I-E^-\phi)^{-1}$ exists and has non-negative terms. (2.N1.1) becomes

$$T = (I-E^-\phi)^{-1}R \qquad (2.N1.4)$$

$(I-E^{-}\phi)^{-1}_{i,j}$ is the fraction of R_i that appears in T_j. Since there is no looping, the terms of $(I-E^{-}\phi)^{-1}$ are no greater than one. We have $(I-E^{-}\phi)_{j,dest}$ equal 0 if $j \neq dest$ and 1 if $j=dest$. Using (2.N1.4) in $F=\phi T$ gives

$$F = \phi(I-E^{-}\phi)^{-1}R \tag{2.N1.5}$$

We now wish to show that $I-\phi E^{-}$ is invertible

$$(I-\phi E^{-})(I+\phi E^{-}+(\phi E^{-})^2+ \ldots +(\phi E^{-})^N)$$

$$= I - (\phi E^{-})^{N+1}$$

$$= I - \phi(E^{-}\phi)^N E^{-}$$

$$= I$$

Therefore $I-\phi E^{-}$ is invertible and

$$(I-\phi E^{-})^{-1} = I + \phi E^{-} + (\phi E^{-})^2 + \ldots + (\phi E^{-})^N \tag{2.N1.6}$$

$$= I + \phi(I+E^{-}\phi+(E^{-}\phi)^2+ \ldots +(E^{-}\phi)^{N-1})E^{-}$$

$$= I + \phi(I-E^{-}\phi)^{-1}E^{-} \tag{2.N1.7}$$

We have the elementary equation

$$\phi(I-E^{-}\phi) = (I-\phi E^{-})\phi$$

Inverting the factors in the parentheses gives

$$(I-\phi E^{-})^{-1}\phi = \phi(I-E^{-}\phi)^{-1} \tag{2.N1.8}$$

Using this in (2.N1.6) gives another expression for F.

$$F = (I-\phi E^-)^{-1}\phi R \qquad (2.N1.9)$$

$(I-\phi E^-)^{-1}_{ij,mn}$ is the fraction of $\phi_{mn}R_m$ that appears in F_{ij}. Therefore, it is non-negative and no greater than one. From (2.N1.7) we have

$$(I-\phi E^-)^{-1}_{ij,mn} = I_{ij,mn} + \phi_{ij}(I-E^-\phi)^{-1}_{i,n} \qquad (2.N1.10)$$

The last term makes sense as all of the flow on (m,n) reaches node n. $(I-E^-\phi)^{-1}$ is the fraction of that flow that reaches node i. ϕ_{ij} is the fraction of this flow that goes on link (i,j). We now develop an inequality using the following property of ϕ (from (2.1.3))

$$\Sigma_j \phi_{ij} = \begin{cases} 1 & \text{if } i \neq dest \\ 0 & \text{if } i = dest \end{cases} \qquad (2.N1.11)$$

$$\begin{aligned} \Sigma_{ij}(I-\phi E^-)^{-1}_{ij,mn} &= 1 + \Sigma_{ij}\phi_{ij}(I-E^-\phi)^{-1}_{i,n} \\ &= 1 + \sum_{i\neq dest} (I-E^-\phi)^{-1}_{i,n} \\ &\leq 1 + \sum_{i\neq dest} 1 \\ &= N \end{aligned} \qquad (2.N1.12)$$

Note 2.N.2. The node distance G has some interesting properties. This is the node distance that was used in [Gallager 77, Bertsekas 78, and Gafni 79]. From (2.2.9) we have

$$G = g\phi(I-E^{-}\phi)^{-1} \tag{2.N2.1}$$

Comparing this with (2.N1.5) shows that the fraction of g_{ij} in G_n is the same as the fraction of R_n in F_{ij}. The most important property of G is GR=gF

$$GR = g\phi(I-E^{-}\phi)^{-1}R = gF \tag{2.N2.2}$$

From remark A.2.1 in appendix A we have the following error bound

$$J(f) - J_{min} \leq gf - \Sigma_k D^k F^k \tag{2.N2.3}$$

where D^k is the shortest distance vector with respect to commodity k and distance g. Since $gf = \Sigma_k gF^k$ we may use (2.N2.2) to get

$$\begin{aligned} J(f) - J_{min} &\leq \Sigma_k G^k R^k - \Sigma_k D^k R^k \\ &= \Sigma_k (G^k - D^k) R^k \end{aligned} \tag{2.N2.4}$$

Let us restrict the routing problem to one commodity and use (2.N1.5) to make J(F) a function of ϕ and R, i.e.

$$J^*(\phi,R) = J(\phi(I-E^{-}\phi)^{-1}R) \tag{2.N2.5}$$

We will differentiate this equation to see how a small change in ϕ and R changes J^*. We will need the differential of $(I-E^{-}\phi)^{-1}$. Let $Y = (I-E^{-}\phi)^{-1}$. Then, $(I-E^{-}\phi)Y = I$. Differentiating this gives $(I-E^{-}\phi)dY - E^{-}d\phi Y = 0$. Rearranging this then gives

$$d(I-E^{-}\phi)^{-1} = dY$$

$$= (I-E^{-}\phi)^{-1}E^{-}d\phi Y$$

$$= (I-E^{-}\phi)^{-1}E^{-}d\phi(I-E^{-}\phi)^{-1}$$

Using the chain rule in (2.N2.5) gives

$$dJ^* = g[d\phi(I-E^{-}\phi)^{-1}R + \phi(I-E^{-}\phi)^{-1}E^{-}d\phi(I-E^{-}\phi)^{-1}R$$

$$+ \phi(I-E^{-}\phi)^{-1}dR] \qquad (2.N2.6)$$

Using (2.N1.4) and (2.N2.1),

$$dJ^* = gd\phi T + GE^{-}d\phi T + GdR \qquad (2.N2.7)$$

We have the constraints,

$$\Sigma_j d\phi_{ij} = 0 \qquad d\phi_{ij} + \phi_{ij} \geq 0 \qquad (i,j)\in L$$

$$R_i + dR_i \geq 0 \text{ for } i\neq dest, dR_{dest} = -\sum_{i\neq dest} dR_i \qquad (2.N2.9)$$

Since $G_{dest}=0$, (2.N2.6) says that if dR_i and dR_{dest} are the only changes in ϕ and R then J* (and J) changes by approximately $G_i dR_i$ where the approximation is better if the change is small.

We end this note with two inequalities. The first is

$$G_i \leq N \max_{mn} g_{mn} \qquad (2.N2.10)$$

This follows from (2.N2.1),

$$G_i = \Sigma_{jk} g_{jk} \phi_{jk} (I-E^- \phi)^{-1}_{j,i}$$

$$\leq \Sigma_{jk} (\max_{mn} g_{mn}) \phi_{jk} (I-E^- \phi)^{-1}_{j,i}$$

$$= (\max_{mn} g_{mn}) \Sigma_j (I-E^- \phi)^{-1}_{j,i}$$

$$\leq N \max_{mn} g_{mn} \qquad (2.N2.11)$$

The other inequality is

$$N(G-D)R \geq (G-D)T \qquad (2.N2.12)$$

To get this inequality we start with (2.2.6) which is repeated here

$$G = (g+GE^-)\phi \qquad (2.N2.13)$$

We have

$$D \leq (g+DE^-)\phi \qquad (2.N2.14)$$

Taking the difference, and then successively multiplying by $E^-\phi$ gives

$$G-D \geq (G-D)E^-\phi$$

$$\geq (G-D)(E^-\phi)^2$$

$$\geq (G-D)(E^-\phi)^3$$

$$\cdots$$

$$\geq (G-D)(E^-\phi)^{N-1}$$

Summing the column of the above chain of inequality and adding G-D = G-D gives

$$N(G-D) \geq (G-D)(I+E^{-}\phi+(E^{-}\phi)^{2} + \ldots + (E^{-}\phi)^{N-1})$$

$$= (G-D)(I-E^{-}\phi)^{-1} \tag{2.N2.15}$$

Since $G_{dest}=0=D_{dest}$ and $(I-E^{-}\phi)^{-1}_{i,dest}=0$ for $i \neq dest$, we can multiply (2.N2.15) by R and preserve the inequality.

$$N(G-D)R \geq (G-D)(I-E^{-}\phi)^{-1}R$$

$$= (G-D)T \tag{2.N2.16}$$

This proves (2.N2.12)

Note 2.N.3. Another possibility for H in algorithm I is the following

$$H^{1/2}_{dest} = 0$$

$$H^{1/2}_{i} = (\Sigma_{j} h_{ij}\phi^{2}_{ij})^{1/2} + \Sigma_{j} H^{1/2}_{j}\phi_{ij} \tag{2.N3.1}$$

The square of $H^{1/2}_{j}$ is used in step IC. This note will prove that this choice of H makes $g\tilde{F} + Ch\tilde{F}^{2}$ non-positive. If the proof of remark 2.2.1 is reviewed it will be seen that it will be enough to show that (2.2.21) still holds. This condition is

$$H_{n} \geq \Sigma_{ij} h_{ij}\phi^{2}_{ij}\theta^{2}_{i,n} \tag{2.N3.2}$$

As in the transformation of G from (2.2.5) to (2.2.10), we have for (2.N3.1)

$$H_n^{1/2} = \Sigma_i (\Sigma_j h_{ij} \phi_{ij}^2)^{1/2} \phi_{i,n} \tag{2.N3.3}$$

Squaring both sides and then reducing the square of the sum to the sum of square gives (2.N3.2).

It is not clear whether this H is better or worse than the H of algorithm I. The one given there was chosen for its simpler computations.

Note 2.N.4. Figure 2.N.4. gives a non-optimal flow for which one iteration of either algorithm of this chapter could fail to make the cost function decrease. However, there is an improvement in ϕ.

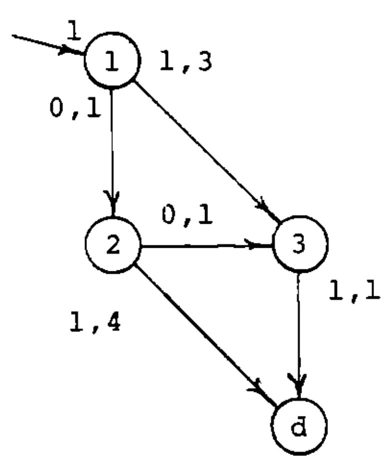

Figure 2.N.4.

Associated with each link (i,j) in the figure is ϕ_{ij}, g_{ij}. There is an input flow only at node 1. It is destined for node d.

In applying one iteration of either algorithm I or II to the figure we have G_2=4, G_3=1, and at node 1, $g_{12}+G_2$ = 5 and $g_{13}+g_3$ = 4. Thus, there is no change in the flow at node 1 or elsewhere. Thus, the cost function does not change even though flow in the figure is not optimal. The optimal flow is on the path (1,2.3,d).

At node 2, $g_{23}+G_3$ = 2 and $g_{2d}+G_d$ = 4 so by either algorithm I or II we have $\tilde{\phi}_{23}$=1. So, we do have an improvement in ϕ and in the next iteration the flow will change and the cost function decrease.

Chapter III.

Partially Distributed Routing

This chapter gives a class of loopfree routing algorithms and shows that each algorithm in this class converges to the optimal flow. Our objective is to provide extensive freedom in choosing the parameters, so that heuristics may be taken advantage of without jeopardizing the convergence to the optimum cost.

The central form of these algorithms is to use a distributed procedure, such as either of the algorithms of chapter two, to determine for each commodity a proposed flow change $\tilde{F}$; A central node then receives the aggregate flow changes $\tilde{f}$ and determines the scale γ that minimizes $J(f+\gamma\tilde{f})$ over $0 \leq \gamma \leq 1$. For each commodity, the new flow is $F+\gamma\tilde{F}$.

Figure 3.1 illustrates a potential problem with this central form. In the figure there is just one commodity, that destined for node d. The flows F_{ij} are given next to the links. ξ and δ are positive. The link costs are

$$J_{ij}(F_{ij}) = \frac{1}{5} \cdot \frac{F_{ij}}{5-F_{ij}}$$

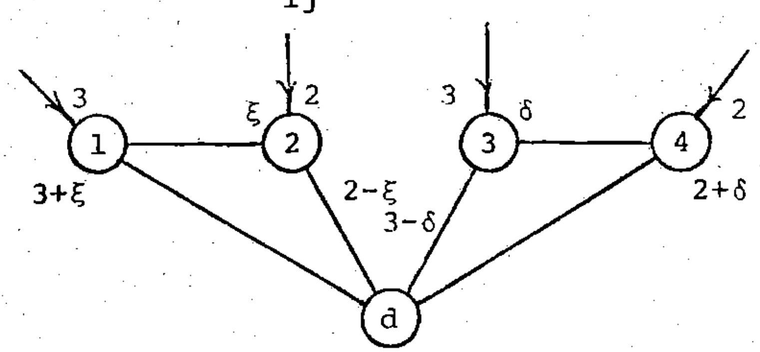

Figure 3.1

The two subnetworks determined by nodes 1,2,d and 3,4,d, respectively, are each the same as that of figure 1.2.1. The situation depicted by nodes 1,2,d might have come about when the input flows were $R_2=3$ and $R_1=2$. Refering to the text associated with figure 1.2.1 we see that the optimal flows are $\delta = .3232$, $\xi = 0$, $F_{1d} = 3-.3232$, $F_{12} = .3232$, and $F_{2d} = 2.3232$. As the routing algorithms (such as those of chapter two) require the flow to be loopfree, node 1 must wait until $\xi=0$ before it can send any flow to node 2. As ξ is on a long path node 2 will set the flow changes $\tilde{F}_{21} = -\xi$ and $\tilde{F}_{2d} = \xi$. In the next iteration ξ will be zero if $\gamma = 1$.

We are interested in routing algorithms that make large flow changes so as to have rapid convergence if convergence exists. These algorithms would over-correct the situation at node 3. Thus if $\delta < .3232$ then $\tilde{F}_{34} > .3232 - \delta$ and if $\delta > .3232$ then $\tilde{F}_{34} < .3232 - \delta$. (We assume that in the presence of the flow change at node 2 the optimal value of δ will never be found.) If the overcorrection is large enough then γ will not be one. γ will be between 0 and 1. In the next iteration δ will be closer to .3232 and ξ will be smaller but not zero. If the overcorrection of δ persists in every iteration then ξ will approach zero as the number of iterations gets large but never become zero. The algorithm would then converge to a non-optimal flow.

The obvious remedy is to make sure overcorrections do not occur but we do not wish to sacrifice rapid convergence. What we will do is have a distributed procedure generate, for each commodity, two flow changes, $\tilde{F}$ and $\bar{F}$, where $\tilde{F}$ is the normal sized flow change and $\bar{F}$ is a small flow change in a good enough direction such that $J(f+\bar{f}) \leq J(f)$. The flow change $\bar{F}$ will be carried out regardless of the scaling. A central node receives the aggregate flow changes $\tilde{f}$ and $\bar{f}$ and determines the scale γ that minimizes $J(f+\bar{f}+\gamma(\tilde{f}-\bar{f}))$ over $0 \leq \gamma \leq 1$. For each commodity the new flow will be $F+\bar{F}+\gamma(\tilde{F}-\bar{F})$. This is the form of the class of routing algorithms given in the next section.

In the example, the algorithm waits until ξ is small enough (this is made specific in the next section) and then sets $F_{21} = -\xi$. In the next iteration ξ will be zero and this will allow node 1 to send flow to node 2.

3.1 A Generalized Algorithm with Scaling

Let f^0 be the initial loopfree flow and $\mathcal{F}$ be the set $\{f \mid J(f) \leq J(f^0), f=\Sigma_k F^k, EF^k=R^k, F^k \geq 0, k\in\mathcal{C}\}$. We assume that on $\mathcal{F}$ the network cost $J = \Sigma_{ij} J_{ij}(f_{ij})$ is twice continuously differentiable with

$$\frac{\partial J_{ij}(f_{ij})}{\partial f_{ij}} > 0 \quad \text{and} \quad \frac{\partial^2 J_{ij}(f_{ij})}{\partial f_{ij}^2} \geq 0 \quad (i,j)\in\mathcal{L} \tag{3.1.1}$$

Let M, B, Λ, and λ be constants satisfying

$$M \geq \max_{f\in\mathcal{F}} \max_{(i,j)\in\mathcal{L}} \left\{ \frac{\partial^2 J_{ij}(f_{ij})}{\partial f_{ij}^2} \right\} \tag{3.1.2}$$

$$M > 0 \tag{3.1.3}$$

$$B = 2MCLN \tag{3.1.4}$$

$$\Lambda \geq \lambda > 0 \tag{3.1.5}$$

(Recall that C is the number of commodities, L the number of links, and N the number of nodes.)

The following outline gives the steps of one iteration. For the first iteration we assume that ϕ is loopfree.

I. Upstream Stage. For each commodity the following steps are done.

A. The destination node sends its neighbors the signal "$W_{dest} = 0$".

B. Each node i waits until it receives the node distance W_j from every j such that $\phi_{ij} > 0$. Then

1. $\mathcal{W}_{ij} = g_{ij}+W_j \quad j:\phi_{ij} > 0$ (g_{ij} was given in (1.1.7).)

2. W_i is chosen arbitrarily from the interval

$$[\min_{j:\phi_{ij}>0} \mathcal{W}_{ij}, \Sigma_j \mathcal{W}_{ij}\phi_{ij}]$$

3. Node i is 'in loop danger' if, for some j, $\phi_{ij} > 0$ and either (a) $W_j > W_i$ or (b) j is in loopdanger.

4. Node i sends W_i and its loopdanger status to its neighbors.

II. Downstream Stage. For each commodity the following steps are done.

A. The set of downhill neighbors Z_i is the set $\{j\}$ such that either (a) $\phi_{ij} > 0$ or (b) $\mathcal{W}_{ij} \leq W_i$ and j is not in loopdanger. If j is a downhill neighbor of i then i is an uphill neighbor of j.

B. Each node i waits until it receives the flow changes $\tilde{F}_{ki}$ and $\overline{F}_{ki}$ from all of its uphill neighbors. Then

1. $\tilde{T}_i = \Sigma_k \tilde{F}_{ki} \qquad \overline{T} = \Sigma_k \overline{F}_{ki}$

2. $\{\mathcal{V}_{ij}\}$ is aribtrarily selected such that (a) $\Lambda \geq \mathcal{V}_{ij}$ and (b)

$$\Sigma_j \mathcal{V}_{ij}\tilde{\phi}^2_{ij} \geq \lambda \Sigma_j \tilde{\phi}^2_{ij}$$

holds for all $\tilde{\phi}$ satisfying $\Sigma_j \tilde{\phi}_{ij} = 0$.

3. $\{\tilde{\phi}_{ij}\}$ is chosen to minimize either

   (a) $\Sigma_j W_{ij} \tilde{\phi}_{ij} + \frac{1}{2} \Sigma_j V_{ij} \tilde{\phi}_{ij}^2 T_i$ or

   (a') $\Sigma_j W_{ij} \tilde{\phi}_{ij} + \frac{1}{2} \Sigma_j V_{ij} \tilde{\phi}_{ij}^2 (T_i + \tilde{T}_i)$

   such that

   (b) $\Sigma_j \tilde{\phi}_{ij} = 0 \quad \phi_{ij} + \tilde{\phi}_{ij} \geq 0$

   (c) $\tilde{\phi}_{ij} = 0$ if $j \notin Z_i$

4. $\{\psi_{ij}\}$ is arbitrarily chosen such that

   (a) $\Sigma_j W_{ij} \psi_{ij} \leq W_i$

   (b) $\psi_{ij} = 0$ if $\phi_{ij} + \tilde{\phi}_{ij} = 0$

5. $\tilde{F}_{ij} = (\phi_{ij} + \tilde{\phi}_{ij})(T_i - \tilde{T}_i^-) + \psi_{ij} \tilde{T}_i^+ - F_{ij}$ (recall that $\tilde{T}_i^- = \max\{0, -\tilde{T}_i\}$ and $\tilde{T}_i^+ = \max\{\tilde{T}_i, 0\}$)

6. $n = \text{arg-min}\ \{W_{ij} | j \in Z_i\}$

7. Any F_{ij} satisfying the following is a 'leak'.

   (a) $0 < F_{ij} \leq B^{-1}(W_{ij} - W_{in})$

   (b) $\phi_{ij} + \tilde{\phi}_{ij} = 0$

8.
$$\bar{\phi}_{ij} = \begin{cases} -\phi_{ij} & \text{if } F_{ij} \text{ is a leak} \\ -\Sigma_{k \neq n} \bar{\phi}_{ik} & \text{if } j = n \\ 0 & \text{otherwise} \end{cases}$$

9. $\overline{F}_{ij} = (\phi_{ij}+\overline{\phi}_{ij})(T_i-\overline{T}_i^-) + \psi_{ij}\overline{T}_i^+ - F_{ij}$

10. Node i sends $\tilde{F}_{ij}$ and $\overline{F}_{ij}$ to each downhill neighbor.

III. Central Stage.

A. When each node i knows the flow changes $\tilde{F}_{ij}$ and $\overline{F}_{ij}$ for all neighbors j and commodities, it computes the aggregate quantities $f_{ij}+\overline{f}_{ij}$ and $\tilde{f}_{ij}-\overline{f}_{ij}$, and sends them to the central node.

B. When the central node receives all of these quantities it determines γ to minimize

$$\Sigma_{ij}J_{ij}(f_{ij}+\overline{f}_{ij}+\gamma(\tilde{f}_{ij}-\overline{f}_{ij}))$$

such that $0 \leq \gamma \leq 1$. This is sent to every node.

C. The new flow is $F^* = F + \overline{F} + \gamma(\tilde{F}-\overline{F})$. The new routing fraction is

$$\phi^*_{ij} = \begin{cases} F^*_{ij}/T^*_i & \text{if} \quad T^*_i > 0 \\ \phi_{ij} + \tilde{\phi}_{ij} & \text{if} \quad T^*_i = 0 \end{cases}$$

Each iteration of the algorithm generates a set of possible feasible flows F^* each depending on the choice of the node distance W, the weight V, and the routing ψ of the node flow increment. Let $A(F)$ be this set after one iteration. $A(F)$ implicitly depends on ϕ, Λ, λ, and B.

The second algorithm of chapter two may be used to generate $\tilde{\phi}$. In this case $\Lambda \geq \frac{1}{2}$ PCLN $\geq \lambda$. The first algorithm of chapter two may also be used if there exists a positive λ

such that

$$CL\Sigma_j (h_{ij}+H_j)\tilde{\phi}_{ij}^2 \geq \lambda\Sigma_j\tilde{\phi}_{ij}^2 \tag{3.1.6}$$

For these algorithms W_i is equal to the maximum value in the interval

$$[\min_{k:\phi_{ik}>0} W_{ik},\ \Sigma_j W_{ij}\ \phi_{ij}]$$

and ψ equals $\phi + \tilde{\phi}$. Algorithm I would minimize IIB3a' while algorithm II minimizes IIB3a.

For this W (= G) and ψ, condition IIB4a is automatically satisfied. To see this note that if IIB3a is minimized then

$$\Sigma_j W_{ij}\tilde{\phi}_{ij} \leq \Sigma_j W_{ij}\tilde{\phi}_{ij} + \frac{1}{2}\ \Sigma_j V_{ij}\tilde{\phi}_{ij}^2 T_i \leq 0$$

If IIB3a' is minimized then

$$\Sigma_j W_{ij}\tilde{\phi}_{ij} \leq \Sigma_j W_{ij}\tilde{\phi}_{ij} + \frac{1}{2}\ \Sigma_j V_{ij}\tilde{\phi}_{ij}^2 (T_i+\tilde{T}_i) \leq 0$$

In either case, $\Sigma_j W_{ij}\tilde{\phi}_{ij} \leq 0$. So

$$\begin{aligned}\Sigma_j W_{ij}\psi_{ij} &= \Sigma_j W_{ij}(\phi_{ij}+\tilde{\phi}_{ij}) \\ &\leq \Sigma_j W_{ij}\phi_{ij} \\ &= W_i \end{aligned} \tag{3.1.7}$$

The watershed distance method used in the previous chapter just prevents loops from developing, which in turn prevents deadlocks from occurring in the algorithm. The loopdanger method of algorithm *A* is another way of doing this. We will show this shortly. The loopdanger method is slightly harder to analyze, gives a more restricted set of downhill neighbors, but also uses boolean numbers "loopdanger status" rather than real numbers S in the communication between nodes. (For that matter the communication of watershed distance could be changed to "S_i is the same as G_i" or "S_i is different from G_i and is ..." This requires using more than just boolean numbers only when $S_i \neq G_i$)

We now show that Z is loopfree. This will avoid a deadlock in the downstream stage and also make ϕ^* loopfree. A loopfree ϕ^* avoids a deadlock in the upstream stage of the next iteration.

We fix our terminology. Node j is a downstream neighbor of i if $\phi_{ij} > 0$. Node j is downstream of i if it is a downstream neighbor of i or if it is downstream of a downstream neighbor of i. Node j is a downhill neighbor of i if $j \in Z_i$. Node j is downhill of i if it is a downhill neighbor of i or if it is downhill of a downhill neighbor of i. Upstream and uphill are the reverse of the downstream and downhill relations.

We assume that ϕ is loopfree. That is, no node is downstream of itself. When a node is in loopdanger its uphill neighbors are just its upstream neighbors. A node in loopdanger is uphill of itself only if it is upstream of itself. But ϕ is loopfree. So, no node in loopdanger is uphill of itself. If j is a downstream neighbor of a node i that is not in loopdanger then $W_j \leq W_i$. If j is a downhill but not downstream neighbor of i then $W_j < g_{ij} + W_j = W_{ij} \leq W_i$. Therefore, if j is a downhill neighbor of i then $W_j \leq W_i$ with equality only if j is a downstream neighbor of i. A node not in loopdanger is downhill of itself only if it is downstream of itself. But ϕ is loopfree. So, no node not in loopdanger is downhill of itself. Thus, altogether, Z is loopfree.

3.2 Convergence

In a long series of remarks we will show that algorithm A converges to the optimal flow. Some of these remarks assume

$$\Lambda \geq B \geq 4\lambda \qquad (3.2.1)$$

even though the remarks could be established with different numerical constants if this were not assumed. From IIB2 one sees that there is no loss in generality in increasing Λ to be greater than B and decreasing λ to be less than B/4.

The first four remarks develop an inequality between $g\tilde{f}$ and $|\tilde{f}|$. This inequality will also show that the directional derivative $g\tilde{f}$ is non-positive.

Remark 3.2.1 $g\tilde{F} \leq W\tilde{\phi}(T-\tilde{T}^-)$

Proof. $g\tilde{F} = \Sigma_{ij} g_{ij}\tilde{F}_{ij}$. Using IB1 in this expression gives,

$$g\tilde{F} = \Sigma_{ij}(W_{ij}-W_j)\tilde{F}_{ij}$$

$$= \Sigma_{ij}W_{ij}\tilde{F}_{ij} - \Sigma_j W_j\tilde{T}_j$$

$$= W\tilde{F} - W\tilde{T} \qquad (3.2.2)$$

Using $\phi T = F$ in IIB5 gives

$$\tilde{F} = \tilde{\phi}(T-\tilde{T}^-) + \psi\tilde{T}^+ - \phi\tilde{T}^- \tag{3.2.3}$$

Using this in (3.2.2) gives

$$g\tilde{F} = W\tilde{\phi}(T-\tilde{T}^-) + W\psi\tilde{T}^+ - W\phi\tilde{T}^- - W\tilde{T}$$

Using $\tilde{T} = \tilde{T}^+ - \tilde{T}^-$ and then rearranging the terms gives

$$g\tilde{F} = W\tilde{\phi}(T-\tilde{T}^-) + (W\psi-W)\tilde{T}^+ + (W-W\phi)\tilde{T}^- \tag{3.2.4}$$

From IB2 we have $W \leq W\phi$. From IIB4a, $W\psi \leq W$. Using both of these in (3.2.4) gives the remark.

Remark 3.2.2. $W\tilde{\phi}(T-\tilde{T}^-) \leq -\lambda|\tilde{\phi}(T-\tilde{T}^-)|^2$

Proof. By using the same argument that led to (2.2.3), if node i minimizes IIB3a then

$$\Sigma_j W_{ij}\tilde{\phi}_{ij} + \Sigma_j V_{ij}\tilde{\phi}^2_{ij}T_i \leq 0 \tag{3.2.5}$$

Alternately, if node i minimizes IIB3a' then

$$\Sigma_j W_{ij}\tilde{\phi}_{ij} + \Sigma_j V_{ij}\tilde{\phi}^2_{ij}(T_i+\tilde{T}_i) \leq 0 \tag{3.2.6}$$

From IIB2b, $\Sigma_j V_{ij}\tilde{\phi}^2_{ij} \geq \lambda\Sigma_j\tilde{\phi}^2_{ij}$. We also have $T_i \geq T_i-\tilde{T}^-_i$ and $T_i + \tilde{T}_i \geq T_i - \tilde{T}^-_i$. So, from the above two cases, (3.2.5) and (3.2.6), we see that

$$\Sigma_j W_{ij}\tilde{\phi}_{ij} + \lambda\Sigma_j\tilde{\phi}^2_{ij}(T_i-\tilde{T}^-_i) \leq 0 \tag{3.2.7}$$

Multiplying this by $T_i-\tilde{T}^-_i$ and then summing over i gives the remark.

Remark 3.2.3 $|\tilde{F}|^2 \leq \frac{1}{2} NL|\tilde{\phi}(T-\tilde{T}^-)|^2$

Proof. Equation (3.2.3) is the same as equation (2.3.6) with ψ in place of ϕ'. If this substitution is maintained in the equations subsequent to (2.3.6) we get the remark from (2.3.24).

Remark 3.2.4. $|\tilde{f}|^2 \leq -\frac{CNL}{2\lambda} g\tilde{f}$

Proof. Combining the previous three remarks,

$$|\tilde{F}|^2 \leq -\frac{NL}{2\lambda} g\tilde{F} \tag{3.2.8}$$

Since $|\tilde{f}|^2 \leq C\Sigma_K|\tilde{F}^K|^2$ and $\Sigma_K g\tilde{F}^K = g\tilde{f}$, the remark follows.

Comment. This remark implies that the directional derivative $g\tilde{f}$ is non-positive.

The following four remarks follow a similar development for $\overline{f}$.

Remark 3.2.5. $g\overline{F} \leq W\overline{\phi}(T-\overline{T}^-)$

Proof. Using $\phi T = F$ in IIB9 gives

$$\overline{F} = \overline{\phi}(T-\overline{T}^-) + \psi\overline{T}^+ - \phi\overline{T}^- \tag{3.2.9}$$

The proof of the remark is the same as for remark 3.2.1 except with $\overline{F}$ in place of $\tilde{F}$ and (3.2.9) in place of (3.2.3).

Remark 3.2.6. $W\overline{\phi}(T-\overline{T}^-) \leq -B\,|\overline{\phi}^-(T-\tilde{T}^-)|^2$

Proof. Let the following be defined.

$$K_i := \min_{j\in Z_i} W_{ij} \tag{3.2.10}$$

From IIB8,

$$\Sigma_j W_{ij}\bar{\phi}_{ij}^{+} = K_i \Sigma_j \bar{\phi}_{ij}^{+} \tag{3.2.11}$$

We have

$$\begin{aligned}\Sigma_j W_{ij}\bar{\phi}_{ij} &= \Sigma_j W_{ij}\bar{\phi}_{ij}^{+} - \Sigma_j W_{ij}\bar{\phi}_{ij}^{-} \\ &= K_i W_j \bar{\phi}_{ij}^{+} - \Sigma_j W_{ij}\bar{\phi}_{ij}^{-} \\ &= K_i W_j \bar{\phi}_{ij}^{-} - \Sigma_j W_{ij}\bar{\phi}_{ij}^{-} \\ &= \Sigma_j (K_i - W_{ij})\bar{\phi}_{ij}^{-} \end{aligned} \tag{3.2.12}$$

If F_{ij} is a leak then, from IIBi, $\bar{\phi}_{ij}^{-} = \phi_{ij}$. In this case, $\bar{\phi}_{ij}^{-}T_i = \phi_{ij}T_i = F_{ij}$. Then with IIB7a, if F_{ij} is a leak

$$\bar{\phi}_{ij}^{-}T_i \leq B^{-1}(W_{ij}-K_i) \tag{3.2.13}$$

If F_{ij} is not a leak then, from IIB8, $\bar{\phi}_{ij}^{-} = 0$ and (3.2.13) still holds. Using (3.2.13) in (3.2.12),

$$\begin{aligned}\Sigma_j W_{ij}\bar{\phi}_{ij} &\leq - B\Sigma_j (\bar{\phi}_{ij}^{-})^2 T_i \\ &\leq - B\Sigma_j (\bar{\phi}_{ij}^{-})^2 (T_i - \bar{T}_i^{-}) \end{aligned} \tag{3.2.14}$$

Multiplying this by $T_i - \bar{T}_i^{-}$ and summing over i gives the remark.

<u>Remark 3.2.7</u> $|\bar{F}|^2 \leq 2NL|\bar{\phi}^{-}(T-\bar{T}^{-})|^2$

Proof. Equation (3.2.9) is the same as equation (2.3.6) except with bar quantities in place of tilde quantities and with ψ in place of ϕ'. If this substitution is maintained in the equations subsequent to (2.3.6) we get the following from (2.3.23).

$$|\overline{F}|^2 \leq \frac{N}{2}\ (\Sigma_{mn}|\overline{\phi}_{mn}|(T_m-\overline{T}_m^{\,-}))^2$$

$$= \frac{N}{2}(2\Sigma_{mn}\overline{\phi}_{mn}^{\,-}(T_m-T_m^{\,-}))^2$$

$$\leq 2NL\Sigma_{mn}(\overline{\phi}_{mn}^{\,-}(T_m-\overline{T}_m^{\,-}))^2$$

$$= 2NL|\overline{\phi}^{\,-}(T-\overline{T}^{\,-})|^2$$

This is the remark.

<u>Remark 3.2.8</u>. $M|\overline{f}|^2 \leq -g\overline{f}$

Proof. Combining the previous three remarks,

$$|\overline{F}|^2 \leq -2NLB^{-1}g\overline{F} \tag{3.2.15}$$

With (3.14), which is B = 2MCLN, we get

$$MC|\overline{F}|^2 \leq -g\overline{F} \tag{3.2.16}$$

Since $|\overline{f}|^2 \leq C\Sigma_k|\overline{F}^k|^2$ and $\Sigma_k g\overline{F}^k = g\overline{f}$, the remark follows.

Comment. This remark implies that the directional derivative $g\overline{f}$ is non-positive.

The next two remarks develop an inequality between the cost change and the directional derivatives.

If $f + \overline{f} + \gamma(\tilde{f}-\overline{f}) \in F$ then with (3.1.2),

$$J(f+\overline{f}+\gamma(\tilde{f}-\overline{f})) - J(f) \leq g(\overline{f}+\gamma(\tilde{f}-\overline{f})) + \frac{1}{2}M|\overline{f}+\gamma(\tilde{f}-\overline{f})|^2 \tag{3.2.17}$$

The following remark sidesteps the F condition.

<u>Remark 3.2.9</u>. For any $\overline{f},\gamma,\tilde{f}$ that makes RHS(3.2.17) non-positive the inequality (3.2.17) holds.

Proof. Let $\hat{f} = \bar{f} + \gamma(\tilde{f}-\bar{f})$. If the remark holds then $J(f+\hat{f}) - J(f) \le 0$. Suppose to the contrary that $J(f+\hat{f}) - J(f) > 0$ and that $\hat{f}$ makes RHS(3.2.17) non-positive, i.e. $g\hat{f} + \frac{1}{2} M|\hat{f}|^2 \le 0$. Because of the former supposition, $\hat{f} \neq 0$. Then $g\hat{f} \le -\frac{1}{2} M|\hat{f}|^2 < 0$. Because $g\hat{f} < 0$ and J is continuous there exists an α, $0 < \alpha < 1$, such that $J(f+\alpha\hat{f}) = J(f)$. Then $f+\alpha\hat{f} \in \mathcal{F}$ and

$$
\begin{aligned}
J(f+\alpha\hat{f}) - J(f) &\le \alpha g\hat{f} + \frac{1}{2} M\alpha^2 |\hat{f}|^2 \\
&< \alpha g\hat{f} + \alpha\frac{1}{2} M|\hat{f}|^2 \\
&= \alpha(g\hat{f} + \frac{1}{2} M|\hat{f}|^2) \\
&\le 0
\end{aligned}
$$

That is, $J(f+\alpha\hat{f}) - J(f) < 0$. This contradicts our selection of α. Therefore, the assumption must be false.

Let the following be defined.

$$\Delta J := \min_{0\le\gamma\le 1} J(f+\bar{f}+\gamma(\tilde{f}-\bar{f})) - J(f) \tag{3.2.18}$$

$$\Delta_1 J := \min_{0\le\gamma\le 1} \text{RHS}(3.2.17)$$

$$= \min_{0\le\gamma\le 1} \{g\bar{f}+\gamma g(\tilde{f}-\bar{f}) + \frac{1}{2} M|\bar{f}+\gamma(\tilde{f}-\bar{f})|^2\} \tag{3.2.19}$$

Algorithm $\mathcal{A}$ generates ΔJ. The above remark implies that if $\Delta_1 J \le 0$ then $\Delta J \le \Delta_1 J$.

<u>Remark 3.2.10</u>. $\Delta J \le \lambda B^{-1}(g\bar{f}+g\tilde{f})$

Proof. Using $|a+b|^2 \leq 2|a|^2 + 2|b|^2$,

$$|\bar{f}+\gamma(\tilde{f}-\bar{f})|^2 = |(1-\gamma)\bar{f}+\gamma\tilde{f}|^2$$

$$\leq 2(1-\gamma)^2|\bar{f}|^2 + 2\gamma|\tilde{f}|^2$$

Multiplying this by M and then using remarks 3.2.4 and 3.2.8,

$$M|\bar{f}+\gamma(\tilde{f}-\bar{f})|^2 \leq - 2(1-\gamma)^2 g\bar{f} - MCNL\lambda^{-1}\gamma^2 g\tilde{f}$$

Using B= 2MCNL (from (3.1.4)),

$$M|\bar{f}+\gamma(\tilde{f}-\bar{f})|^2 \leq - 2(1-\gamma)^2 g\bar{f} - B(2\lambda)^{-1}\gamma^2 g\tilde{f} \tag{3.2.20}$$

From (3.2.19),

$$\Delta_1 J \leq g\bar{f} + \gamma g\tilde{f} - \gamma g\bar{f} + \frac{1}{2} M|\bar{f}+\gamma(\tilde{f}-\bar{f})|^2$$

Using (3.2.20),

$$\Delta_1 J \leq g\bar{f} + \gamma g\tilde{f} - \gamma g\bar{f} - (1-\gamma)^2 g\bar{f} - B(4\lambda)^{-1}\gamma^2 g\tilde{f}$$

Expanding $(1-\gamma)^2$ and simplifying,

$$\Delta_1 J \leq \gamma g\bar{f} + \gamma g\tilde{f} - \gamma^2 g\bar{f} - B(4\lambda)^{-1}\gamma^2 g\tilde{f}$$

From (3.2.1), $B(4\lambda)^{-1} \geq 1$. Thus,

$$\Delta_1 J \leq \gamma g\bar{f} + \gamma g\tilde{f} - B(4\lambda)^{-1}\gamma^2 g\bar{f} - B(4\lambda)^{-1}\gamma^2 g\tilde{f}$$

$$= (\gamma - \frac{B}{4\lambda}\gamma^2)(g\bar{f}+g\tilde{f}) \tag{3.2.21}$$

The RHS above is minimized over γ at $\gamma = 2\lambda/B$. This γ is positive and less than one. Using it in the above gives

$$\Delta_1 J \le \lambda B^{-1}(g\bar{f}+g\tilde{f}) \tag{3.2.22}$$

Since the directional derivatives, $g\bar{f}$ and $g\tilde{f}$, are non-positive, $\Delta_1 J$ is non-positive. Thus, $\Delta J \le \Delta_1 J$ and the remark follows.

The next remark says that the flow change is bounded by the cost change.

Remark 3.2.11. $M|\bar{f}+\gamma(\tilde{f}-\bar{f})|^2 \le -\frac{1}{2}(B/\lambda)^2 \Delta J$

Proof. Using $(1-\gamma)^2 \le 1 \le B(4\lambda)^{-1}$ and $\gamma^2 \le 1$ in (3.2.20),

$$M|\bar{f} + \gamma(\tilde{f}-\bar{f})|^2 \le - B(2\lambda)^{-1}(g\bar{f}+g\tilde{f}) \tag{3.2.23}$$

Using the previous remark in the above gives the current remark.

We repeat the definition of K_i (from (3.2.10)) and make two more definitions.

$$K_i := \min_{j\in Z_i} W_{ij} \tag{3.2.24}$$

$$U_i := \max_{j\in Z_i} \{W_{ij} | \phi_{ij}+\bar{\phi}_{ij} > 0\} \tag{3.2.25}$$

$$U_i^! := \max_{j\in Z_i} \{W_{ij} | \phi_{ij}+\tilde{\phi}_{ij} > 0\} \tag{3.2.26}$$

At the end of the iteration we have

$$\max_{j\in Z_i} \{W_{ij} | \phi^*_{ij} > 0\} = \begin{cases} U_i^! & \text{if} \quad \gamma=1 \text{ or } T_i^*=0 \\ U_i & \text{if} \quad \gamma<1 \text{ and } T_i^*\neq 0 \end{cases} \tag{3.2.27}$$

Since $\phi_{ij} + \tilde{\phi}_{ij} > 0$ implies $\phi_{ij} + \bar{\phi}_{ij} > 0$, we have $U_i^! \le U_i$. In the following four remarks it will be shown that $|U-K|$ is bounded by the cost change.

<u>Remark 3.2.12</u>. $|U'-K|^2 \leq 2\Lambda^2|\tilde{\phi}(T+\tilde{T}^+)|^2$

Proof. We consider first the case of node i minimizing IIB3a which is

$$\Sigma_j W_{ij}\tilde{\phi}_{ij} + \frac{1}{2}\Sigma_j V_{ij}\tilde{\phi}^2_{ij}T_i \tag{3.2.28}$$

As in (2.2.2), any change from the optimal $\tilde{\phi}$ to, say, $\tilde{\psi}$ is a non-descent change in (3.2.37). Thus,

$$\Sigma_j[W_{ij} + V_{ij}\tilde{\phi}_{ij}T_i](\tilde{\psi}_{ij}-\tilde{\phi}_{ij}) \geq 0 \tag{3.2.29}$$

The bracketed quantity is the gradient of (3.2.28). Let k be the arg-min of (3.2.24) and u be the arg-max of (3.2.26). Let $\epsilon = \phi_{iu} + \tilde{\phi}_{iu}$. By the definition of u, ϵ is positive. Let $\tilde{\psi}$ be the following

$$\tilde{\psi}_{ij} = \begin{cases} \tilde{\phi}_{iu} - \epsilon & \text{if } j=u \\ \tilde{\phi}_{ij} & \text{if } u \neq j \neq k \\ \tilde{\phi}_{ik} + \epsilon & \text{if } j=k \end{cases} \tag{3.2.30}$$

(3.2.29) becomes

$$[U'_i + V_{iu}\tilde{\phi}_{iu}T_i](-\epsilon) + [K_i + V_{ik}\tilde{\phi}_{ik}T_i]\epsilon \geq 0$$

Dividing by ϵ and using IIB2a,

$$- U'_i + \Lambda|\tilde{\phi}_{iu}|T_i + K_i + \Lambda|\tilde{\phi}_{ik}|T_i \geq 0$$

Rearranging and squaring,

$$(U'_i-K_i)^2 \leq \Lambda^2(|\tilde{\phi}_{iu}| + |\tilde{\phi}_{ik}|)^2 T_i^{\,2}$$

$$\leq 2\Lambda^2(\tilde{\phi}_{iu}^2+\tilde{\phi}_{ik}^2)T_i^2$$

$$\leq 2\Lambda^2|\tilde{\phi}_i|^2T_i^2 \tag{3.2.31}$$

For the case of node i minimizing IIB3a' the corresponding argument yields

$$(U_i'-K_i)^2 \leq 2\Lambda^2|\tilde{\phi}_i|^2(T_i+\tilde{T}_i)^2 \tag{3.2.32}$$

In either case,

$$(U_i'-K_i)^2 \leq 2\Lambda^2|\tilde{\phi}_i|^2 \max\{T_i^2,(T_i+\tilde{T}_i)^2\}$$

$$= 2\Lambda^2|\tilde{\phi}_i|^2(T_i+\tilde{T}_i^+)^2$$

$$= 2\Lambda^2|\tilde{\phi}_i(T_i+\tilde{T}_i^+)|^2 \tag{3.2.33}$$

Summing over i gives the remark.

<u>Remark 3.2.13</u>. $|U-K|^2 \leq 2\Lambda^2|\tilde{\phi}(T+\tilde{T}^+)|^2$

Proof. Let m be the arg-max of (3.2.25). We have $\phi_{im}+\bar{\phi}_{im} > 0$.

Case 1, $U_i' < U_i$. In this case $\phi_{im} + \tilde{\phi}_{im} = 0$. Since F_{im} is not a leak it does not satisfy condition IIB7a. Thus, $F_{im} \geq B^{-1}(U_i-K_i)$.

$$\Sigma_j\tilde{\phi}_{ij}^2T_i^2 \geq \tilde{\phi}_{im}^2T_i^2$$

$$= \phi_{im}^2T_i^2$$

$$= F_{im}^2 \tag{3.2.34}$$

$$\geq B^{-2}(U_i-K_i)^2 \tag{3.2.35}$$

Since $B^2 \leq \Lambda^2 \leq 2\Lambda^2$.

$$2\Lambda^2 \Sigma_j \tilde{\phi}_{ij}^2 T_i^2 \geq (U_i - K_i)^2 \tag{3.2.36}$$

Case 2, $U_i' = U_i$. From (3.2.33), the above inequality is again satisfied.

Summing (3.2.36) over i then gives the remark.

Remark 3.2.14. $|\tilde{\phi}(T+\tilde{T}^+)|^2 \leq NL|\tilde{\phi}(T-\tilde{T}^-)|^2$

Proof. We first derive an expression for $\tilde{T}^-$. We have $\tilde{T}^- \leq E^-\tilde{F}^-$. With (2.3.7),

$$\tilde{T}^- \leq E^-\tilde{\phi}^-(T-\tilde{T}^-) + E^-\phi\tilde{T}^- \tag{3.2.37}$$

Substracting the last term from both sides and then multiplying by θ_N (defined in (2.2.8)),

$$\tilde{T}^- \leq \theta_N E^-\tilde{\phi}^-(T-\tilde{T}^-) \tag{3.2.38}$$

Elementwise, this is

$$\begin{aligned} \tilde{T}_i^- &\leq \Sigma_{mn}\theta_{i,n}\tilde{\phi}_{mn}^-(T_m-\tilde{T}_m^-) \\ &\leq \Sigma_{mn}\tilde{\phi}_{mn}^-(T_m-\tilde{T}_m^-) \\ &= \frac{1}{2}\Sigma_{mn}|\tilde{\phi}_{mn}|(T_m-\tilde{T}_m^-) \end{aligned} \tag{3.2.39}$$

We next derive a similar expression for $\tilde{T}^+$. Here we start with $\tilde{T}^+ \leq E^-\tilde{F}^+$ and (2.3.17) with ψ in place of ϕ'.

$$\tilde{T}^+ \leq E^-\tilde{\phi}^+(T-\tilde{T}^-) + E^-\psi\tilde{T}^+ \tag{3.2.40}$$

We define the following,

$$\theta_N'' := (I-E^-\psi)^{-1} \tag{3.2.41}$$

This has properties similar to θ_N including $0 \leq \theta''_{i,n} \leq 1$. Subtracting the last term of (3.2.40) from both sides and multiplying θ_N'' gives

$$\tilde{T}^+ \leq \theta_N'' E^- \tilde{\phi}^+ (T-\tilde{T}^-) \tag{3.2.42}$$

Elementwise, this is

$$\begin{aligned} \tilde{T}_i^+ &\leq \Sigma_{mn} \theta''_{i,n} \tilde{\phi}^+_{mn} (T_m-\tilde{T}_m^-) \\ &\leq \Sigma_{mn} \tilde{\phi}^+_{mn} (T_m-\tilde{T}_m^-) \\ &= \frac{1}{2} \Sigma_{mn} |\tilde{\phi}_{mn}| (T_m-\tilde{T}_m^-) \end{aligned} \tag{3.2.43}$$

We now prove the remark. We have the identity,

$$|\tilde{\phi}_{ij}|(T_i+\tilde{T}_i^+) = |\tilde{\phi}_{ij}|(T_i-\tilde{T}_i^-) + |\tilde{\phi}_{ij}|(\tilde{T}_i^-+\tilde{T}_i^+)$$

Both $\tilde{T}_i^-$ and $\tilde{T}_i^+$ cannot be positive. Using this fact and (3.2.39) and (3.2.43) in the last term,

$$|\tilde{\phi}_{ij}|(T_i+\tilde{T}_i^+) \leq |\tilde{\phi}_{ij}|(T_i-\tilde{T}_i^-) + \frac{1}{2}|\tilde{\phi}_{ij}|\Sigma_{mn}|\tilde{\phi}_{mn}|(T_m-\tilde{T}_m^-)$$

Since $\tilde{\phi}_{mn}=0$ when m=dest, there are no more than L terms in the RHS. Squaring both sides and using Cauchy's inequality gives

$$(\tilde{\phi}_{ij}(T_i+\tilde{T}_i^+))^2 \leq L(\tilde{\phi}_{ij}(T_i-\tilde{T}_i^-))^2 + \frac{1}{2}L\tilde{\phi}_{ij}^2|\tilde{\phi}(T-\tilde{T}^-)|^2$$

Summing this over (i,j),

$$|\tilde{\phi}(T+\tilde{T}^+)|^2 \leq L|\tilde{\phi}(T-\tilde{T}^-)|^2 + \frac{1}{2} L\Sigma_{ij}\tilde{\phi}_{ij}^2|\tilde{\phi}(T-\tilde{T}^-)|^2 \tag{3.2.44}$$

$\Sigma_j \tilde{\phi}_{ij}^2$ is no greater than 2 and is zero when i=dest. Thus, $\frac{1}{2} \Sigma_{ij} \tilde{\phi}_{ij}^2 \leq (N-1)$. The remark follows.

Remark 3.2.15. For every commodity, $|U-K|^2 \leq -2BNL\Lambda^2\lambda^{-2}\Delta J$.

Proof. Combining remarks 3.2.13, 3.2.14, 3.2.2, and 3.2.1.,

$$|U-K|^2 \leq -2NL\Lambda^2\lambda^{-1}\, g\tilde{F} \tag{3.2.45}$$

This holds for every commodity k so $g\tilde{F}^k$ are all non-positive. Thus, $g\tilde{F} \geq \Sigma_k g\tilde{F}^k = g\tilde{f}$. The above continues.

$$\begin{aligned} |U-K|^2 &\leq -2NL\Lambda^2\lambda^{-1} g\tilde{f} \\ &\leq -2NL\Lambda^2\lambda^{-1}(g\tilde{f}+g\bar{f}) \end{aligned} \tag{3.2.46}$$

Remark 3.2.10 then gives the remark.

Let {f(0), f(1),...} be a sequence of flows generated by A, i.e., $f(m+1) \in A(f(m))$. Let $\Delta J(m) = J(f(m+1)) - J(f(m))$.

Remark 3.2.16. $\Delta J(m)$ approaches zero as m becomes large.

Proof. The set F is closed and bounded. On this set there is a minimum cost

$$J_{min} := \min_{f \in F} J(f) \tag{3.2.47}$$

With remark 3.2.15, {J(0), J(1), ...} is a monotone decreasing sequence bounded by J_{min}. Therefore, the sequence converges to some number and the remark follows.

Let f^c be a cluster point in the sequence {f(m)}. It exists because F is closed and bounded. Let $\{f(m_i)\}$ be a subsequence of $\{f(m)\}$ that converges to f^c. Since remarks 3.2.16 and 3.2.11 say that $|f(m+1) - f(m)|$ goes to zero, the sequence $\{f(m_i+1)\}$ also converges to f^c. Likewise for $\{f(m_i+2)\}, \ldots, \{f(m_i+N-1)\}$.

Let g^c be the gradient of $J(f^c)$. Let D^c be the shortest distance wrt g^c to a particular destination. In the next five remarks we will show that there exists an m' such that $m_i > m'$ and $D_j^c > g_{jk}^c + D_k^c$ imply $F_{jk}(m_i+N-1) = 0$. Let S^c be the node neighborhood of D^c, i.e.

$$(i,j) \in S^c \text{ if and only if } D_i^c = g_{ij}^c + D_j^c \tag{3.2.48}$$

Let s be the smallest miss from the shortest distance, i.e.

$$s = \min \{g_{ij}^c + D_j^c - D_i^c \mid (i,j) \notin S^c\} \tag{3.2.49}$$

Let g_{min} be defined by

$$g_{min} = \min_{f \in F} \min_{(i,j) \in L} \frac{\partial J_{ij}(f_{ij})}{\partial f_{ij}} \tag{3.2.50}$$

Because of (3.1.1) and the fact that F is closed and bounded, $g_{min} > 0$. Let ϵ be any number satisfying

$$\min\{s, g_{min}\} > \epsilon > 0 \tag{3.2.51}$$

Let m' be such that for $m_i > m'$ and for $n = m_i, m_i+1, \ldots, m_i+N-1$ the following holds.

$$|g(n) - g^c| \leq \frac{\epsilon}{4N} \tag{3.2.52a}$$

$$|U(n) - K(n)| \leq \frac{\epsilon}{4} \tag{3.2.52b}$$

(3.2.52b) is valid because of remarks 3.2.15 and 3.2.16. By expanding g as a function of f we have

$$|g(m)-g^c| \leq M|f(m)-f^c| \tag{3.2.53}$$

(3.2.52a) is then valid because, for each ℓ, $\{f(m_i+\ell)\}$ converges to f^c.

In the following we work with a flow F such that (3.2.52a &b) holds. We say that a node i in F is tight if (1) $\phi_{ij} > 0$ implies $(i,j)\in S^c$ and (2) $D_j^c < D_i^c$ implies j is tight. The destination node is trivially tight.

Remark 3.2.17. If the present iteration satisfies (3.2.52a) and if $(i,j)\in S^c$ and j is tight then

$$W_{ij} \leq D_i^c + \frac{\epsilon}{4}$$

Proof. Suppose the if clauses hold.

$$\begin{aligned} W_{ij} &= g_{ij} + W_j \\ &\leq g_{ij} + \Sigma_k W_{jk}\phi_{jk} \end{aligned} \tag{3.2.54}$$

Let $m = \text{arg-max}\{W_{jk}|\phi_{jk} > 0\}$. Since node j is tight, $(j,m)\in S^c$. Therefore, $D_m^c < D_j^c$ and m is tight. (3.2.54) continues

$$\begin{aligned} W_{ij} &\leq g_{ij} + W_{im} \\ &\leq g_{ij} + g_{jm} + W_m \\ &\leq g_{ij} + g_{jm} + g_{mn} + \cdots \end{aligned} \tag{3.2.55}$$

where $(i,j),(j,m),(m,n),\ldots\in S^c$

$$\begin{aligned}
\mathcal{W}_{ij} &\le g_{ij} - g_{ij}^c + g_{ij}^c + g_{jm} - g_{jm}^c + g_{jm}^c + \dots \\
&\le |g_{ij}-g_{ij}^c| + g_{ij}^c + |g_{jm}-g_{jm}^c| + g_{jm}^c + \dots \\
&\le |g-g^c| + g_{ij}^c + |g-g^c| + g_{jm}^c + \dots \\
&\le (N-1)|g-g^c| + D_i^c \\
&\le \frac{\epsilon}{4} + D_i^c
\end{aligned} \tag{3.2.56}$$

The last inequality used (3.2.52a).

<u>Remark 3.2.18</u>. If the present iteration satisfies (3.2.52a) and if $(i,j) \notin \mathcal{S}^c$ then

$$\mathcal{W}_{ij} \ge D_i^c + \frac{3\epsilon}{4}$$

Proof.

$$\begin{aligned}
\mathcal{W}_{ij} &= g_{ij} + W_j \\
&\ge g_{ij} + K_j
\end{aligned} \tag{3.2.57}$$

Let $\mathcal{W}_{jm} = K_m$. Then

$$\begin{aligned}
\mathcal{W}_{ij} &\ge g_{ij} + \mathcal{W}_{jm} \\
&\ge g_{ij} + g_{jm} + K_m \\
&\ge g_{ij} + g_{jm} + g_{mn} + \dots \\
&\ge g_{ij} - g_{ij}^c + g_{ij}^c + g_{jm} - g_{jm}^c + g_{jm}^c + \dots \\
&\ge -|g_{ij}-g_{ij}^c| + g_{ij}^c - |g_{jm}-g_{jm}^c| + g_{jm}^c - \dots \\
&\ge -|g-g^c| + g_{ij}^c - |g-g^c| + g_{jm}^c -
\end{aligned} \tag{3.2.58}$$

$$W_{ij} \geq -(N-1)|g-g^c| + D_i^c + s \qquad (3.2.59)$$

D_i^c+s appears here because $(i,j)\notin S^c$ so the expression $g_{ij}^c+g_{jm}^c+\ldots$ is not along a shortest path in D_i^c. Using (3.2.51) and (3.2.52a) leads to the remark.

Remark 3.2.19. If the present iteration satisfies (3.2.52a) then the tight nodes are not in loopdanger.

Proof. We proceed by induction. The destination node is trivially tight and not in loopdanger. Suppose that i is tight and that $D_j < D_i$ implies j is not in loopdanger. With an expansion of K_i similar to (3.2.58) except with $(i,j),(j,m),\ldots\in S^c$ we have

$$\begin{aligned} W_i &\geq K_i \\ &\geq D_i^c - \frac{\epsilon}{4} \qquad (3.2.60) \end{aligned}$$

Since i is tight, $\phi_{ij} > 0$ implies that j is tight. With remark 3.2.17,

$$\begin{aligned} W_j &\leq \Sigma_m W_{jm}\phi_{jm} \\ &\leq D_j^c + \frac{\epsilon}{4} \qquad (3.2.61) \end{aligned}$$

Then,

$$\begin{aligned} W_i - W_j &\geq D_i^c - D_j^c - \frac{\epsilon}{2} \\ &= g_{ij}^c - \frac{\epsilon}{2} \\ &\geq \frac{\epsilon}{2} \\ &> 0 \qquad (3.2.62) \end{aligned}$$

(3.2.62) used (3.2.51). IB3 then says that node i is not in loopdanger.

Remark 3.2.20. If the present iteration satisfies (3.2.52a) then the tight nodes remain tight in the next iteration.

Proof. We proceed by induction. The destination node trivially remains tight in the next iteration. Suppose that node i is tight and that each node j with $D_j^c < D_i^c$ remains tight in the next iteration. With remark 3.2.17 we have

$$W_i \le \Sigma_m W_{im}\phi_{im}$$

$$\le D_i^c + \frac{\epsilon}{4} \qquad (3.2.63)$$

If a node j with $\phi_{ij} = 0$ enters Z_i then from IIAb, $W_{ij} \le W_i$. Using (3.2.63), $W_{ij} \le D_i^c + \epsilon/4$. Remark 3.2.18 then implies $(i,j)\in S^c$. Therefore, $j\in Z_i$ implies $(i,j)\in S_i^c$. If $\phi_{ij}^* > 0$ then $j\in Z_i$ and $(i,j)\in S_i^c$ and $D_j^c < D_i^c$. From our supposition j remains tight in the next iteration. Therefore, node i remains tight.

Remark 3.2.21. If the present iteration satisfies (3.2.52a&b) and not all nodes are tight then a node will become tight in the next iteration.

Proof. Let i = arg-min $\{D_j^c \mid j \text{ is not tight}\}$. Note that with this selection if $(i,m)\in S^c$ then m is tight. We first show that Z_i contains some m for which $(i,m)\in S^c$. If $\phi_{im} > 0$ for such an m then $m\in Z_i$. Thus, suppose that $(i,m)\notin S^c$ for all m such that $\phi_{im} > 0$. Then remark 3.2.18 and step IB2 say

$$W_i \geq D_i^c + \frac{3\epsilon}{4} \tag{3.2.64}$$

If m is such that $(i,m) \in S^c$ then m is tight and remark 3.2.17 says

$$W_{im} \leq D_i^c + \frac{\epsilon}{4} \tag{3.2.65}$$

Thus, $W_{im} \leq W_i$. Since remark 3.2.19 says that node m is not in loopdanger, m enters Z_i.

The first part of this proof has shown that Z_i has a node m such that $(i,m) \in S^c$. Thus, remark 3.2.17 says

$$K_i \leq D_i^c + \frac{\epsilon}{4} \tag{3.2.66}$$

With (3.2.52b),

$$\begin{aligned} U_i &= |U_i - K_i| + K_i \\ &\leq |U-K| + D_i^c + \frac{\epsilon}{4} \\ &\leq D_i^c + \frac{\epsilon}{2} \end{aligned} \tag{3.2.67}$$

A review of IIIC and (3.2.25) shows that

$$\max_j \{W_{ij} | \phi_{ij}^* > 0\} \leq D_i^c + \frac{\epsilon}{2} \tag{3.2.68}$$

Remark 3.2.18 then implies that if $\phi_{ij}^* > 0$ then $(i,j) \in S^c$. The previous remark says that tight nodes remain tight in the next iteration. Thus, node i will be tight in the next iteration.

Remark 3.2.21. In each flow in $\{f(m_i+N-1)\}$, $m_i > m'$, all nodes are tight.

Proof. With the previous remark at least one node (the destination) is tight in $f(m_i)$, two nodes in $f(m_i+1), \ldots,$ all nodes in $f(m_i-N-1)$.

The next two remarks prove the convergence of $\{J(m)\}$ to J_{min}.

Remark 3.2.22. For any cluster point f^c of $\{f(m)\}$ and for any ϵ satisfying (3.2.51) there is a subsequence $\{f(m_j)\}$ such that every commodity in each $f(m_j)$ satisfies

$$gF - DR \leq Nr\epsilon$$

where $r = \Sigma_k \Sigma_{i \neq k} R_i^k$.

Proof. We use $\{m_j\} = \{m_i+N | m_i > m'\}$. D_i has an expansion in g,

$$\begin{aligned} D_i &= g_{ij} + g_{jk} + \ldots \\ &= g_{ij} - g_{ij}^c + g_{ij}^c + g_{jk} - g_{jk}^c + g_{jk}^c + \ldots \\ &\geq -|g-g^c| + g_{ij}^c - |g-g^c| + g_{jk}^c - \ldots \\ &\geq D_i^c - (N-1)|g-g^c| \\ &\geq D_i^c - \frac{\epsilon}{4} \end{aligned} \tag{3.2.69}$$

Since i is tight by the previous remark, we have

$$\begin{aligned} \Sigma_j W_{ij} \phi_{ij} &\leq D_i^c + \frac{\epsilon}{4} \\ &\leq D_i + \frac{\epsilon}{2} \end{aligned} \tag{3.2.70}$$

The last inequality used (3.2.69). We rewrite the above as

$$W\phi_i - D_i \leq \frac{\epsilon}{2} \leq \epsilon$$

Multiplying this by T_i,

$$W\phi_i T_i - D_i T_i \leq \epsilon T_i \leq \epsilon r$$

Summing over i,

$$W\phi T - DT \leq Nr\epsilon \qquad (3.2.71)$$

We have

$$\begin{aligned} W\phi T - DT &= WF - DT \\ &= \Sigma_{ij}(g_{ij}+w_j)F_{ij} - DT \\ &= gF + \Sigma_j W_j (T_j - R_j) - DT \\ &\geq gF + D(T-R) - DT \\ &= gF - DR \qquad (3.2.72) \end{aligned}$$

This with (3.2.71) gives the remark.

Remark 3.2.23. The cluster points of $\{f(m)\}$ are optimal flows.

Proof. Since ϵ is arbitrarily small in the previous remark, we have for each cluster point f^c and every commodity

$$g^c F^c - D^c R = 0 \qquad (3.2.73)$$

Remark A.1.5 in appendix A then says that f^c is optimal.

Note 3.N.1. This note shows that if $\lambda \geq \frac{1}{4}B = \frac{1}{2}$ MCLN then the algorithm can safely use $F+\tilde{F}$ for the new flow F^* bypassing $\bar{F}$ and the calculation of γ. Let this simplified algorithm be $A'(f)$ and let $\{f(m)\}'$ be the sequence of flows generated by A', i.e. $f(m-1) \in A'(f(m))$.

Remark 3.N1.1. If $\lambda \geq \frac{1}{2}$ MCLN then the cluster points of $\{f(m)\}'$ are optimal flows.

Proof. Remarks 3.2.1-4 still hold. Remark 3.2.4 simplifies into

$$M|\tilde{f}|^2 \leq - g\tilde{f} \tag{3.N1.1}$$

Remark 3.2.9 still holds. Let

$$\Delta'J = J(f+\tilde{f}) - J(f) \tag{3.N1.2}$$

$$\Delta_1'J = g\tilde{f} + \frac{1}{2} M|\tilde{f}|^2 \tag{3.N1.3}$$

Algorithm A' generates $\Delta'J$. Using $\gamma=1$, remark 3.2.9 says that if $\Delta_1'J$ is non-positive then $\Delta'J \leq \Delta_1'J$. In place of remark 3.2.10 we will show that $\Delta'J \leq \frac{1}{2} g\tilde{f}$. (3.N1.1) says that gf is non-positive. Using (3.N1.1) in (3.N1.3) gives $\Delta_1'J = \frac{1}{2} g\tilde{f}$. Thus, $\Delta_1'J$ is non-positive. Thus,

$$\Delta'J \leq \frac{1}{2} g\tilde{f} \tag{3.N1.4}$$

In place of remark 3.2.11 we combine (3.N1.1) and (3.N1.4) to get

$$M|\tilde{f}|^2 \leq -2\Delta'J \tag{3.N1.5}$$

At the end of the iteration we will have

$$\max_j\{W_{ij}\,|\,\phi^*_{ij} > 0\} = U'_i \tag{3.N1.6}$$

So U' instead of U is the quantity of interest in the middle part of the proof. Remarks 3.2.12 and 3.2.14 still hold and they combine with remarks 3.2.2 and 3.2.1 and (3.N1.4) to get

$$|U'-K|^2 \le -4NL\Lambda^2\lambda^{-1}\Delta' J \tag{3.N1.7}$$

in place of remark 3.2.15. Remarks 3.2.16-23 still hold with $\Delta' J$ in place of ΔJ, $\{f(m)\}'$ in place of $\{f(m)\}$ and U' in place of U. This proves the remark.

<u>Note 3.N.2</u>. This note describes Gallager's [77] algorithm. It can be used to generate $\tilde{\phi}$ in the routing outline of this chapter. In this algorithm the node distance is $W_i = \Sigma_j W_{ij}\phi_{ij}$. Let β be a positive scalar. For each node i the following steps are done.

$$n = \text{arg-min}\{W_{ij}\,|\,j\in Z_i\} \tag{3.N2.1}$$

$$\tilde{\phi}_{ij} = \begin{cases} -\min\left\{\phi_{ij}, \dfrac{W_{ij}-W_{in}}{\beta T_i}\right\} & \text{if } j\neq n, j\in Z_i \\ -\Sigma_{k\neq n}\tilde{\phi}_{ik} & \text{if } j=n \end{cases} \tag{3.N2.2}$$

$$\tilde{F}_{ij} = (\phi_{ij}+\tilde{\phi}_{ij})(T_i+\tilde{T}_i) - F_{ij} \qquad j\in {}_i \tag{3.N2.3}$$

To match the routing outline to this algorithm we would have $\psi = \phi+\tilde{\phi}$ and

$$V_{ij} = \begin{cases} 0 & \text{if } j=n \\ \beta & \text{if } j\neq n \end{cases} \tag{3.N2.4}$$

We require $\Lambda \geq \beta$. With the following we will show that $\lambda \leq \beta/N$ satisfies IIB2b.

$$\Sigma_j \tilde{\phi}^2_{ij} = \Sigma_j (\tilde{\phi}^-_{ij})^2 + \Sigma_j (\tilde{\phi}^+_{ij})^2$$

$$\leq \Sigma_j (\tilde{\phi}^-_{ij})^2 + (\Sigma_j \tilde{\phi}^+_{ij})^2$$

$$= \Sigma_j (\tilde{\phi}^-_{ij})^2 + (\Sigma_j \tilde{\phi}^-_{ij})^2$$

$$\leq \Sigma_j (\tilde{\phi}^-_{ij})^2 + (N-1)\Sigma_j (\tilde{\phi}^-_{ij})^2$$

$$= N\Sigma_j (\tilde{\phi}^-_{ij})^2 \qquad (3.N2.5)$$

$$\Sigma_j V_{ij}\tilde{\phi}^2_{ij} = \beta\Sigma_j (\tilde{\phi}^-_{ij})^2 \geq \frac{\beta}{N} \Sigma_j \tilde{\phi}^2_{ij} \geq \lambda\Sigma_j \tilde{\phi}^2_{ij} \qquad (3.N2.6)$$

Appendix A

Dual of the Routing Problem

The first section of this appendix derives several conditions under which the flow is optimal. The second section gives a bound on the error $J(f) - J_{min}$. All of the results given here can be found in the literature. They are included here for the sake of completeness.

A.1. Linear Programming Application

In section 3.1 the cost function is defined on the set $\mathcal{F} = \{f|J(f) \leq J(f^0),\ f = \Sigma_k F^k,\ EF^k = R^k,\ F^k \geq 0,\ k\in\mathcal{C}\}$. By (3.1.1) J is convex and has positive partial derivatives on $\mathcal{F}$. In this section it will be convenient to assume that $J_{min} < J(f^0)$. (If this is not so then every flow in $\mathcal{F}$ is automatically optimal.) We recast the routing problem as

$$\begin{aligned} &\text{minimize } J(f') \\ &\text{Such that } J(f') < J(f^0) \\ &\qquad f' = \Sigma_k F'^k \\ &\qquad EF'^k = R^k \\ &\qquad F'^k \geq 0 \\ &\qquad k\in\mathcal{C} \end{aligned} \tag{A.1.1}$$

Let g be the gradient at f. Its elements are positive.

$$g = \frac{\partial J(f)}{\partial f} \tag{A.1.2}$$

Remark A.1.2. If f is optimal in (A.1.1) then each commodity F is optimal in the following problem. If every commodity F is optimal in the following problem then f is optimal in (A.1.1).

$$\begin{aligned} &\text{minimize } gF' \\ &\text{such that } EF' = R \\ &\qquad\qquad F' \geq 0 \end{aligned} \tag{A.1.3}$$

Proof. Suppose that f is optimal in (A.1.1). For any F' satisfying the constraints of (A.1.3) there exists an $\epsilon' > 0$ such that the flow $f + \epsilon(F'-F)$ satisfies the constraints of (A.1.1) for all ϵ, $0\leq\epsilon\leq\epsilon'$. Because f is optimal in (A.1.1),

$$\left.\frac{\partial J(f+\epsilon(F'-F))}{\partial \epsilon}\right|_{\epsilon = 0} \geq 0 \tag{A.1.4}$$

That is, $g(F'-F) \geq 0$. This says that for any F', $gF \leq gF'$. Thus, F is optimal in (A.1.3).

Now suppose that every commodity F is optimal in (A.1.3). Let f' be any object that satisfies the constraints of (A.1.1). F' satisfies the constraints of (A.1.3) and since F is optimal there, $gF \leq gF'$. Since J is convex,

$$\begin{aligned} J(F') &\geq J(f) + g(f'-f) \\ &= J(f) + \Sigma_k g(F'^k - F^k) \\ &\geq J(f) \end{aligned} \tag{A.1.5}$$

Therefore, f is optimal in (A.1.1).

The dual problem of (A.1.3) is

maximize $\Pi'R$

such that $\Pi'E \leq g$ (A.1.6)

Since $R_{dest} = -\Sigma_{i \neq dest} R_i$ we have $\Pi'R = \Sigma_i \Pi_i' R_i = \Sigma_{i \neq dest}$ $(\Pi_i' - \Pi_{dest}') R_i$. With the translation $\Pi_i = \Pi_i' - \Pi_{dest}'$ the dual problem becomes

maximize ΠR

such that $\Pi_i - \Pi_j \leq g_{ij}$ $(i,j) \in L$ (A.1.7)

$\Pi_{dest} = 0$

Let D be the shortest distance wrt g, i.e.

$$D_{dest} = 0, \ D_i = \min_j \{g_{ij} + D_j\} \quad \text{(A.1.8)}$$

Remark A.1.2. For any R, the shortest distance D is optimal in the dual problem.

Proof. By an induction, we will show that $D \geq \Pi$. Since R_i is non-negative for $i \neq$ dest this will give $DR \geq D\Pi$ and prove the remark. We have $D_{dest} = \Pi_{dest}$. Now suppose that $D_j \geq \Pi_j$ for all j such that $D_j < D_i$. Let $m = \text{arg-}\min_j \{g_{ij} + D_j\}$. Then

$$\begin{aligned} D_i &= g_{ij} + D_m \\ &\geq g_{ij} + \Pi_m \\ &\geq \Pi_i \end{aligned} \quad \text{(A.1.9)}$$

This completes the induction.

Linear programming [e.g., Luenberger 73] says: (I) If one of the problems, (A.1.3) or (A.1.6), has an optimal solution then both problems have an optimal solution with the same optimal cost. (II) If problems (A.1.3) and (A.1.6) have a solution with the same optimal cost than that solution is optimal. This and the preceding remark gives

Remark A.1.3. F is optimal in (A.1.3) if and only if $gF = DR$. From $R = EF$ comes $gF = DR = DEF$ or $(g-DE)F = 0$. Since $g - DE \geq 0$, this gives

Remark A.1.4. F is optimal in (A.1.3) if and only if $D_i < g_{ij} + D_j$ implies $F_{ij} = 0$

Using the above two remarks with remark A.1.1 gives

Remark A.1.5. The following are equivalent

(i) f is optimal in (A.1.1)

(ii) For every flow f' and every commodity, $gF \leq gF'$

(iii) For every commodity, $gF = DR$

(iv) For every commodity, $D_i < g_{ij} + D_j$ implies $F_{ij} = 0$.

The following remark gives a property of D.

Remark A.1.6. For any F' satisfying the constraints of (A.1.3), $gF' \geq DR$.

Proof. From the constraints of (A.1.6) we have $DE \leq g$. Since F' is non-negative $DEF' \leq gF'$. Since $EF' = R$, the remark follows.

A.2 Error Bound

Remark A.2.1. $J(f) - J_{min} \le gf - \Sigma_k D^k R^k$

Proof. Let f* be the optimal flow. Since J is convex,

$$\begin{aligned} J_{min} = J(f^*) &\ge J(f) + g(f^*-f) \\ &= J(f) + \Sigma_k g F^{*k} - gf \\ &= J(f) + \Sigma_k D^k R^k - gf \end{aligned} \qquad (A.2.1)$$

The last inequality used remark A.1.6. In practice this bound is loose. We might have a flow whose cost agrees with the optimal cost to four significant digits but the error bound will confirm only the first two digits.

Appendix B

Routing Samples

In this appendix, four algorithms are tried on three different networks. In all of the algorithms the proposed flow change was scaled down to minimize the network cost as in step III of the algorithm given in section 3.1. The four algorithms were numbered in the order of their computational complexity. Algorithm 1 is the algorithm given in section 1.2. Algorithm 2 is Gallager's algorithm given in note 3.N.2 We used $\beta = 1$. Algorithm 3 is the algorithm of section 2.3 with the coefficient of the quadratic term in (2.3.1) reduced to 1/2. Algorithm 4 is the algorithm of section 2.2 with the coefficient of the quadratic term in (2.2.1) also reduced to 1/2.

The following is the link cost function that was used. It is a standard mean delay formula which was redefined for $f_{ij} > .999c_{ij}$ so as to facilitate the loading of congested networks. It is twice continuously differentiable.

$$J_{ij}(f_{ij}) = \begin{cases} \dfrac{1}{r} \cdot \dfrac{f_{ij}}{c_{ij} - f_{ij}} & \text{if } f_{ij} \le .999c_{ij} \\[2ex] \dfrac{1}{r} \left(3000 + 3\text{x}10^6 \dfrac{f_{ij}-c_{ij}}{c_{ij}} + 10^9 \left(\dfrac{f_{ij}-c_{ij}}{c_{ij}} \right)^2 \right) & \text{if } f_{ij} > .999c_{ij} \end{cases} \tag{B.1}$$

The first network is given in Figure B.1. Uniform link capacities were used. $c_{ij} = 5$. The input rate R_i^k were generated by uniform random numbers in the interval [0,1] and then held constant. The network was loaded by putting the destination into an empty queue. The front end of the queue was then continually serviced by checking whether all of its neighbors had been enqueued yet. If neighbor m had not been enqueued then it was enqueued and ϕ_{mi} set to one.

In this network the flow branches into at most two parts. Consequently, algorithms 2 and 3 are basically the same. At $\beta = 2$ they would have performed exactly the same. Table B.1 showed what happened.

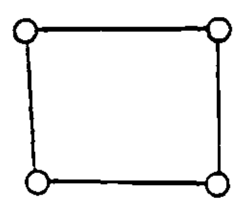

Figure B.1. Four nodes. Eight links.

	Algorithm			
Iteration	1	2	3	4
1	.3709	.3691	.3757	.3686
2	.3697	.3674	.3702	.3674
3	.3681	.3673	.3683	.3673
4	.3677		.3676	
5	.3675		.3674	
6	.3675		.3673	
7	.3674			
8	.3673			

Table B.1 Mean Delay. N = 4, L = 8.
Initial mean delay = .3969

To see what would happen at higher loads the input rate was multiplied by 2.5. Table B.2 gives the result. We stopped algorithm 1 in the tenth iteration. In network routing it is the first few iterations of an algorithm that most interests us.

	1	2	3	4
1	.8829	.8829	.8829	1.3556
2	.8629	.8523	.8341	.9393
3	.8300	.8209	.8091	.8123
4	.8174	.8124	.8032	.8001
5	.8120	.8047	.8006	.7999
6	.8089	.8018	.8001	
7	.8070	.8002	.8000	
8	.8021	.8000	.7999	
9	.8012	.7999		
10	.8009			

Table B.2. Mean Delay. High Loading. $\underline{N} = 4$, $L = 8$.
Initial mean delay = 3.8×10^5

The next network that was tried is given in the following figure. The link capacities were 10. The input rate was determined in the same way as for the first network.

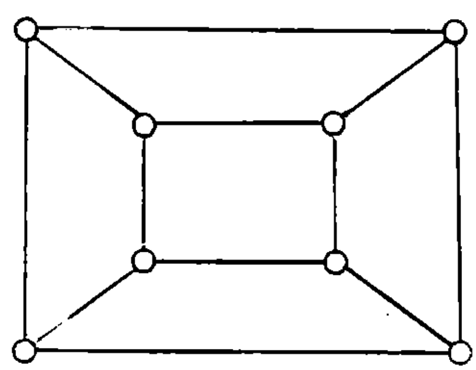

Figure B.2. Eight nodes. 24 links.

With this network we examined another variable. From equation (3.2.4) it is seen that sending the positive node flow changes down the shortest path steepens the descent direction. It also tends to concentrate the positive flow changes onto a few paths, thus, increasing the directional second derivative. We call this diversion of the positive node flow changes down the shortest path splitting. Let ψ represent the routing down the shortest path and let $\tilde{\phi}$ be the routing change generated by the algorithm. Without splitting we have $\tilde{F} = (\phi+\tilde{\phi})(T+\tilde{T}) - F$. With splitting, $\tilde{F} = (\phi+\tilde{\phi})(T-\tilde{T}^-) + \psi\tilde{T}^+ - F$.

This node flow splitting has no effect on algorithm 1. If the other algorithms used the splitting then we called them 2S or 3S or 4S. We did not examine this variable in the first network as those nodes two hops away from destination had no change in their node flow.

Tables B.3 and B.4 show the results. In Table B.4 the input rates were multiplied by 2.5.

	Algorithm						
Iteration	1	2	2S	3	3S	4	4S
1	.2316	.2375	.2353	.2441	.2423	.2316	.2306
2	.2301	.2323	.2312	.2382	.2365	.2296	.2295
3	.2296	.2305	.2301	.2350	.2336	.2294	.2294
4	.2294	.2298	.2297	.2330	.2320		
5		.2296	.2295	.2318	.2310		
6		.2295	.2294	.2310	.2305		
7		.2294		.2305	.2301		
8				.2301	.2299		
9				.2299	.2297		
10				.2297	.2296		

Table B.3. Mean Delay. N = 8, L = 24.
Initial mean delay = .2663

Iteration	Algorithm 1	2	2S	3	3S	4	4S
1	.4760	.4591	.4568	.5526	.4883	.6180	.5374
2	.4389	.4301	.4309	.4588	.4431	.4566	.4444
3	.4359	.4268	.4271	.4367	.4299	.4306	.4292
4	.4346	.4263	.4263	.4297	.4271	.4270	.4260
5	.4335	.4260	.4260	.4274	.4263	.4262	.4258
6	.4331	.4259	.4259	.4265	.4260	.4260	.4258
7	.4326	.4258	.4258	.4261	.4259	.4258	.4257
8	.4323	.4258	.4258	.4259	.4258	.4258	
9	.4321	.4257	.4259	.4258	.4257	.4257	
10	.4319			.4258			

Table B.4. Mean Delay. N = 8, L = 24.
Initial mean delay = 2.5×10^6

For the final network we took the following abstraction of an ARPANET topology [Kleinrock 76, p. 308]. We set the link capacities to 10 and since the topology was non-symmetric we let the input rate $R_i^k = .2$. Table B.5 gives the result.

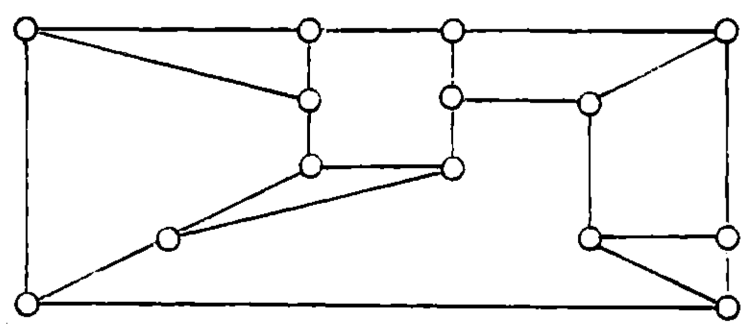

Figure B.3. 14 nodes. 42 links

	1	2	2S	3	3S	4	4S
1	.9546	.9524	.9526	.9508	.9515	.9500	.9509
2	.9417	.9382	.9384	.9375	.9378	.9362	.9364
3	.9400	.9125	.9159	.9161	.9156	.9270	.9248
4	.9345	.9108	.9125	.9136	.9130	.9224	.9197
5	.9312	.9104	.9113	.9127	.9116	.9135	.9119
6	.9261	.9102	.9108	.9120	.9110	.9122	.9112
7	.9233	.9101	.9106	.9116	.9105	.9105	.9107
8	.9218	.9100	.9104	.9112	.9102	.9103	.9105
9	.9206	.9099	.9102	.9109	.9099	.9102	.9103
10	.9191	.9099	.9101	.9107	.9098	.9101	.9102

Table B.5. Mean Delay. N = 14, L = 42.
Initial mean delay = 1.2045

In the third network there were no 3-way flow branching and about 17 2-way flow branchings.

The strongest suggestion that these tables make is that algorithm 1 should not be used. In tables B.4 and B.5 what algorithm 1 reached in ten iterations most of the other algorithms reached in three iterations. Splitting the node flow is of less significance than the difference between algorithms 2 and 3 which itself is not big.

References

1. C. Agnew, "On Quadratic Adaptive Routing Algorithms," Communications of the ACM, Vol. 19, No. 1, pp. 18-22, 1976.

2. D.P. Bertsekas, "Algorithms for Optimal Routing of Flow in Networks," Coordinated Science Laboratory Working Paper, University of Illinois at Champaign-Urbana, June 1978.

3. D.G. Cantor and M. Gerla, "Optimal Routing in a Packet Switched Computer Network," IEEE Trans. Computers, Vol. C-23, pp. 1062-1069, Oct. 1974.

4. H. Frank and W. Chou, "Routing in Computer Networks," Networks, Vol. 1, pp. 99-122, 1971.

5. L. Fratta, M. Gerla, L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," Networks, Vol. 3, pp. 97-133, 1973.

6. R. Gallager, "A Minimal Delay Routing Algorithm Using Distributed Computation," IEEE Trans. Communications, Vol. COM-25, No. 1, pp. 73-85, Jan. 1977.

7. E. M. Gafni, Convergence of a Routing Algorithm, LIDS-R-907, M.I.T., May 1979.

8. L. Kleinrock, Queueing Systems, Vol. II, Wiley-Interscience, 1976.

9. D.G. Luenberger, Introduction to Linear and Non-Linear Programming, Addison-Wesley, 1973.

10. J.M. McQuillan, I. Richer, and E.C. Rosen, "The New Routing Algorithm for ARPANET," IEEE Trans. Commun., Vol. Com-28, pp. 711-719, May 1980.

11. H. Rudin, "On Routing and Delta Routing," IEEE Trans. Commun., Vol. Com-24, pp. 43-59, Jan. 76.

12. H. Rudin and H. Mueller, "Dynamic Routing and Flow Control," IEEE Trans. Commun., Vol. Com-28, pp. 1030-1039, July 1980.

13. M. Schwartz and C.K. Cheung, "The Gradient Projection Algorithm for Multiple Routing in Message-Switched Networks," IEEE Trans. Commun., Vol. Com-24, pp. 449-456, April 1976.

14. M. Schwartz and T.E. Stern, "Routing Techniques Used in Computer Communication Networks," IEEE Trans. Commun., Vol. Com-28, pp. 539-552, April 1980.