

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A089840	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Class of Optimal Routing Algorithms for Communication Networks	5. TYPE OF REPORT & PERIOD COVERED Technical	6. PERFORMING ORG. REPORT NUMBER LIDS-P-1042 ✓
7. AUTHOR(s) Dimitri P. Bertsekas	8. CONTRACT OR GRANT NUMBER(s) ARPA Order No. 3045/5-7-75 ONR/N00014-75-C-1183 ✓	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 5T10 ONR Identifying No. 049-383
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Laboratory for Information and Decision Systems Cambridge, MA 02139	11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE September 1980
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	15. SECURITY CLASS. (of this report) Unclassified	13. NUMBER OF PAGES 16
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES ICCC 80 Atlanta, Ga., Oct. 1980		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) THIS REPORT		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) We describe an algorithm for minimum delay routing in a communication network. During the algorithm each node maintains a list of paths along which it sends traffic to each destination together with a list of the fractions of total traffic that are sent along these paths. At each iteration a minimum marginal delay path to each destination is computed and added to the current list if not already there. Simultaneously the corresponding fractions are updated in a way that reduces average delay per message. The algorithm is		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

80 10 1 024

AD A089840

DDC FILE COPY

DTIC
SELF
OCT 2 1980
C

capable of employing second derivatives of link delay functions thereby providing automatic scaling with respect to traffic input level. It can be implemented in both a distributed and a centralized manner, and it can be shown to converge to an optimal routing at a linear rate.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

ICCC 80
Atlanta, Ga., Oct. 1980

11 Sept 1980

12 17

14 LIDS-P-1042

6 A CLASS OF OPTIMAL ROUTING ALGORITHMS
FOR COMMUNICATION NETWORKS.

9 *Technical rept.*
by

10 Dimitri P. Bertsekas

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Mass. 02139

† Work supported by Grants ONR N00014-75-C-1183, and NSF ENG-7906332.

15
ARPA order 3/45

410950
MTD

ABSTRACT

We describe an algorithm for minimum delay routing in a communication network. During the algorithm each node maintains a list of paths along which it sends traffic to each destination together with a list of the fractions of total traffic that are sent along these paths. At each iteration a minimum marginal delay path to each destination is computed and added to the current list if not already there. Simultaneously the corresponding fractions are updated in a way that reduces average delay per message. The algorithm is capable of employing second derivatives of link delay functions thereby providing automatic scaling with respect to traffic input level. It can be implemented in both a distributed and a centralized manner, and it can be shown to converge to an optimal routing at a linear rate.

I. INTRODUCTION

We consider the problem of optimal quasistatic routing of data in a communication network with slowly varying input traffic statistics. The type of physical situation underlying the problem is the one described in Gallager [1] and we refer to that paper for further discussion. We describe an algorithm for solving the corresponding optimization problem in either a distributed or a centralized manner. The algorithm is similar to Gallager's method [1] and its generalized versions [2] in that it relates to the gradient projection method for nonlinear programming (Goldstein [3], Levitin-Poljak [4] - see Bertsekas [2] for a related discussion). The algorithm given here is different primarily in that it operates in the space of path flows rather than in the space of link flows. Furthermore it utilizes a shortest path computation to obtain a search direction rather than an upstream summation of link marginal delays. As a result the amount of computation per iteration is typically smaller for our algorithm. Our algorithm is also well suited for virtual circuit networks since it operates directly in terms of the quantities of interest - the path flows. In this respect it is similar to a routing algorithm by Segall [5]. The algorithm bears also a close relationship with an algorithm developed in a different context by Aashtiani [6] which utilizes second derivatives. Aashtiani's algorithm need not converge to an optimal routing but, for problems where it does converge, the corresponding computational results have been very encouraging.

Our algorithm is also similar to the flow deviation method (Fratta, Gerla, and Kleinrock [7]) and the extremal flow method (Cantor and Gerla [8]), in that at each iteration it involves the same type of shortest

path computation. However, the first of these methods is characterized by sublinear and hence slow convergence rate ([9], [10]), while the second requires the solution of a nonlinear programming problem at each iteration. By contrast the convergence rate of our method is linear, while the computational requirements per iteration are rather modest. Furthermore, within the context of our method it is possible to employ second derivatives in a manner that is reminiscent of Newton's method. Computational results for related methods that employ second derivatives [11], [12] have been very favorable and this provides grounds for optimism that the same will be true for corresponding versions of our method. An additional benefit of the use of second derivatives is that it provides automatic stepsize scaling with respect to level of traffic input as has been shown computationally within the context of generalized versions of Gallager's method [11]. This advantage is of crucial importance if the algorithm is operated in a distributed manner.

We consider a network consisting of N nodes denoted by $1, 2, \dots, N$ and a set of directed links denoted by L . We denote by (i, ℓ) the link from node i to node ℓ , and assume that the network is connected in the sense that for any two nodes m, n there is a directed path from m to n . We consider the following multicommodity flow problem in the vector of total link flows $f = \{f_{i\ell} \mid (i, \ell) \in L, j = 1, \dots, N\}$:

$$\begin{aligned}
 (1) \quad & \text{minimize } D(f) \\
 & \text{subject to } \sum_{\ell \in O(i)} f_{i\ell}(j) - \sum_{m \in I(i)} f_{mi}(j) = r_i(j), \quad \forall i, j=1, \dots, N \quad i \neq j \\
 & f_{i\ell}(j) \geq 0, \quad \forall (i, \ell) \in L, \quad i, j=1, \dots, N \\
 & f_{j\ell}(j) = 0 \quad \forall (j, \ell) \in L, \quad j = 1, \dots, N.
 \end{aligned}$$

Here $f_{i\ell}(j)$ is the flow on link (i, ℓ) destined for node j , $f_{i\ell} = \sum_{j=1}^N f_{i\ell}(j)$ is the total flow on link (i, ℓ) , $O(i)$ and $I(i)$ are the sets of nodes ℓ for which

$(i,l) \in L$ and $(l,i) \in L$ respectively, and, for $i \neq j$, $r_i(j)$ is a given traffic input at node i destined for j . Each link (i,l) has associated with it a number C_{il} , referred to as the capacity of the link which is assumed positive or $+\infty$. The standing assumptions throughout the paper are:

- a) $r_i(j) \geq 0, \quad \forall i, j = 1, \dots, N, \quad i \neq j$.
- b) The real valued function D is defined on the set

$$(2) \quad S = \{f \mid 0 \leq f_{il} < C_{il}, \quad \forall (i,l) \in L\}$$

and is convex and twice continuously differentiable. Furthermore $\lim_{f_{il} \rightarrow C_{il}} D(f) = +\infty$, $\frac{\partial D(f)}{\partial f_{il}} > 0$, and the Hessian matrix of D is positive definite for all $f \in S$ and $(i,l) \in L$.

Problem (1) is formulated in the space of link flows. We consider an equivalent formulation in the space of path flows. Consider the set of all active origin-destination (OD) pairs

$$A = \{(i,j) \mid r_i(j) > 0, \quad i, j = 1, \dots, N\} .$$

For each $a = (i,j) \in A$ we also write $r_i(j) = r_a$. For each OD pair

$a = (i,j) \in A$ let P_a be the set of directed paths with no repeated nodes originating at i and terminating at j . For $a = (i,j) \in A$ and a path $p \in P_a$ we denote by h_p the flow originating at i destined for j and travelling along the path p . Let $h = \{h_p \mid p \in P_a, \quad a \in A\}$ be the vector of path flows and H be the set of all feasible path flow vectors

$$(3) \quad H = \{h \mid h_p \geq 0, \quad \sum_{p \in P_a} h_p = r_a \quad \forall p \in P_a, \quad a \in A\} .$$

For each $h \in H$ there is a corresponding flow vector f satisfying the constraints

of problem (1) and related to h linearly by means of the equation

$$(4) \quad f_{i\ell} = \sum_{a \in A} \sum_{p \in P_a} \delta_p(i,\ell) h_p, \quad \forall (i,\ell) \in L$$

when $\delta_p(i,\ell) = 1$ if the path p contains the link (i,ℓ) and $\delta_p(i,\ell) = 0$ otherwise. Let us write (4) as

$$(5) \quad f = Eh,$$

where E is the appropriate matrix, and let us consider the function

$$(6) \quad J(h) = D(Eh),$$

and the problem

$$(7) \quad \begin{array}{l} \text{minimize } J(h) \\ \text{subject to } h \in H. \end{array}$$

It is easy to show that problem (7) is equivalent to problem (1) in the sense that an optimal solution h^* of problem (7) yields an optimal solution $\{f_{i\ell}^*(j) \mid (i,\ell) \in L, j = 1, \dots, N\}$ of (1) via the relation

$$f_{i\ell}^*(j) = \sum_{\substack{i \\ (i,j) \in A}} \sum_{p \in P_{(i,j)}} \delta_p(i,\ell) h_p^*.$$

This is based on the assumption $\frac{\partial D}{\partial f_{i\ell}} > 0$ which precludes the existence of optimal sets of flows that contain loops.

The algorithm of this paper solves numerically problem (7) but its definition, as well as its convergence and rate of convergence properties depend crucially on assumption (b) made earlier regarding the function D . In the next section we describe the algorithm for the case where there is a single OD pair. The generalization to the case of several OD pairs and the corresponding convergence and rate of convergence results are given in Section 3. In Section 4 we describe briefly how the algorithm can be operated in a distributed manner.

Proofs of various statements and results of this paper are generally omitted due to space limitations. However, the reader who is familiar with recent work on optimal routing should be able to convince himself of the validity of the algorithm and the corresponding convergence results. References where appropriate are provided as an aid in this respect.

2. The Algorithm for a Single OD Pair

Consider the special case of problem (7) where there is a single OD pair $(i,j) = a$. Problem (7) is written then as

$$(8) \quad \begin{aligned} & \text{minimize } J(h) = D(Eh) \\ & \text{subject to } \sum_{p \in P_a} h_p = r_a \\ & \quad \quad \quad h_p \geq 0, \quad \forall p \in P_a \end{aligned}$$

The algorithm is initiated with a subset of paths $P_a^0 \subset P_a$, and a set of path flows h^0 which may be nonzero only on the paths in P_a^0 , i.e.,

$$(9) \quad h_p^0 = 0 \quad \text{if } p \notin P_a^0,$$

and are such that the corresponding set of total link flows $f^0 = E h^0$ belongs to the set S of (2), i.e.

$$(10) \quad f^0 \in S.$$

At the k th iteration we have a subset of paths $P_a^k \subset P_a$ and corresponding sets of path flows h^k and total link flows f^k satisfying

$$(11) \quad h_p^k = 0 \quad \text{if } p \notin P_a^k, \quad f^k \in S.$$

We compute a shortest path $p^k \in P_a$ by means of a shortest path algorithm using as length of each link $(\ell, m) \in L$ the partial derivative $\frac{\partial D(f^k)}{\partial f_{\ell m}}$.

We set

$$(12) \quad P_a^{k+1} = P_a^k \cup \{p^k\}.$$

(Note here that p^k may already belong to P_a^k). We then solve the quadratic programming problem

$$(13) \quad \begin{aligned} & \text{minimize} \quad \sum_{p \in P_a^{k+1}} \{d_p^k (h_p - h_p^k) + \frac{1}{2} \mu_p^k (h_p - h_p^k)^2\} \\ & \text{subject to} \quad \sum_{p \in P_a^{k+1}} h_p = r_a, \quad h_p \geq 0, \quad \forall p \in P_a^{k+1}, \end{aligned}$$

where for each $p \in P_a^{k+1}$, d_p^k is the total length of the path p with respect to link lengths $\frac{\partial D(f^k)}{\partial f_{\ell m}}$, i.e.

$$(14) \quad d_p^k = \sum_{(\ell, m) \in L} \delta_p(\ell, m) \frac{\partial D(f^k)}{\partial f_{\ell m}},$$

and μ_p^k is a positive scalar. If \bar{h}_p^k is the solution of problem (13) we set for $p \in P_a$

$$(15) \quad h_p^{k+1} = \begin{cases} \bar{h}_p^k & \text{if } p \in P_a^{k+1} \\ 0 & \text{if } p \notin P_a^{k+1} \end{cases}.$$

In the way of explanation we mention that there holds

$$d_p^k = \frac{\partial J(h^k)}{\partial h_p}, \quad \forall p \in P_a^{k+1}$$

[cf. (4), (5), (6) and (14)]. As a result the subproblem (13) may be viewed as a quadratic approximation of the original problem (8) and the algorithm may be viewed as a version of the Goldstein-Levitin-Poljak gradient projection method (see [2]) - the only different being that problem (13) involves only the paths in P_a^{k+1} rather than the entire set of paths P_a . Note that problem (13) involves a single equality constraint and can be solved by essentially analytical means (compare with [2]). It is possible to show by using rather standard

nonlinear programming arguments that there exists a scalar $\mu > 0$ such that if

$$(16) \quad \mu \leq \mu_p^k, \quad \forall p \in P_a^{k+1}, \quad k = 0, 1, \dots$$

and the sequences $\{\mu_p^k\}$ are all bounded, then the generated sequence $\{f^k\}$ satisfies $\{f^k\} \subset S$ and converges to the unique optimal solution of problem (1).

A choice of μ_p^k based on second derivatives is given by

$$\mu_p^k = \eta \frac{\partial^2 J(h^k)}{\partial h_p^2} = \eta \sum_{(m,\ell) \in L} \frac{\partial^2 D(f^k)}{\partial f_{m\ell}^2} \delta_p^{(m,\ell)}$$

where η is a positive scalar. When $\eta = 1$ the resulting algorithm represents a diagonal approximation to Newton's method. Algorithms of this type with η near unity have been shown computationally to be quite successful in a related context [11], [12]. Another possible choice is

$$\mu_p^k = \sum_{(m,\ell) \in L} \eta_{m\ell}^k \frac{\partial^2 D(f^k)}{\partial f_{m\ell}^2} \delta_p^{(m,\ell)}$$

where $\eta_{m\ell}^k$ is the number of paths in P_a^{k+1} that contain link (m,ℓ) .

With this choice it may be shown that the objective function of problem (13) is an upper bound to the one corresponding to Newton's method applied to problem (8). This in turn can be used to show that if f^k is near the optimum, then the k th iteration leads to guaranteed reduction of the objective function.

The algorithm just described does not involve a search for an appropriate stepsize along the descent direction $(\bar{h}^k - h^k)$, and as such it is more suitable for distributed operation. When the algorithm is operated in a centralized manner it is possible to incorporate a linear

search of the type that is usual in many nonlinear programming algorithms thereby guaranteeing a reduction of the objective function value at each iteration (compare with [2]).

3. The Algorithm for Several OD Pairs

In the general case where there are several OD pairs the algorithm consists of several simultaneous OD pair iterations of the type described in the previous section. The set of OD pairs A is partitioned in a union of disjoint subsets A_1, A_2, \dots, A_n where n is some positive integer. The algorithm operates in cycles of n iterations. In the first iteration of each cycle a single iteration of the algorithm of the previous section is carried out simultaneously for all OD pairs $a \in A_1$. In the second iteration of the cycle the same thing is done for all OD pairs in A_2 , and so on until the OD pairs in A_n are iterated upon. At this point the n -iteration cycle is completed and a new cycle begins with the OD pairs in A_1 .

Among the possible partitions of the set A the two extreme cases are when $n = 1$, $A_1 = A$, and when A_i consists of a single OD pair. Other possibilities of interest are when $n = N$ and A_i consists of all OD pairs corresponding to the same origin i or the same destination i .

Any one of these algorithms can be shown to generate sequences $\{f^k\}$ that converge to the unique optimal solution of problem (1) under the assumption that the sequences $\{\mu_p^k\}$ are bounded and satisfy (16) for a sufficiently large lower bound μ .

The proof of this fact is similar but actually simpler than Gafni's convergence proof [13] for a related algorithm that operates in the space

of link flows. Regarding rate of convergence it is possible to show for the case where $n = 1$, $A_1 = A$ that $\{f^k\}$ converges to the optimal solution at a linear rate. A proof of this fact for the case where μ_p^k is a constant may be found in Bertsekas and Gafni [14]. It is our conjecture that the convergence rate is linear even when $n > 1$ but this remains to be shown.

4. Operating the Algorithm in a Distributed Manner

It is possible to distribute the computation involved in each iteration of the algorithm among the nodes of the network. Each node i measures $r_i(j)$ for all destinations j and receives periodically the values of the total flows f_{lm} of all links (l,m) . Node i can then execute the portion of the iteration of the algorithm that corresponds to OD pairs for which i is the origin. This involves computation of shortest paths from i to all destinations and an adjustment of the flows of the currently active paths according to (13), (15). Each node i is also involved in monitoring the average link flows of outgoing links $f_{i\ell}$. The values of these flows are broadcast to all other nodes at the beginning of each iteration by either a flooding scheme, or by means of a spanning tree.

A distributed algorithm of the type just described resembles the current ARPANET routing algorithm [15] in that information providing a length for each link is broadcast throughout the network, each node computes shortest paths from itself to each destination on the basis of these lengths, and shifts flow to the shortest paths. In the ARPANET algorithm all flow is shifted to the shortest path at each iteration. In our algorithm a portion of the flow is retained in previous shortest

paths and, if this is done properly, the resulting flow pattern converges asymptotically to the optimal. By contrast an algorithm of the ARPANET type cannot provide optimal routing since it precludes the possibility of sending data along more than one paths for any single OD pair.

A certain amount of synchronization of link flow broadcasts is necessary in our algorithm. It would be interesting to know what would happen if these broadcasts and the corresponding shortest path computations and flow adjustments were carried out asynchronously as in the ARPANET algorithm. Such an algorithm would certainly offer important practical advantages but the associated questions of convergence are quite difficult and as yet unexplored.

References

- [1] Gallager, R.G., "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Trans. on Communications, Vol. COM-25, 1978, pp.73-85.
- [2] Bertsekas, D.P., "Algorithms for Nonlinear Multicommodity Network Flow Problems", International Symposium on Systems Optimization and Analysis, A Bensoussan and J.L. Lions (eds.), Springer-Verlag, 1979, pp.210-224.
- [3] Goldstein, A.A., "Convex Programming in Hilbert Space", Bull. Amer. Math. Soc., Vol. 70, 1964, pp. 709-710.
- [4] Levitin, E.S., and Poljak, B.T., "Constrained Minimization Problems", U.S.S.R. Comput. Math. Math. Phys., Vol. 6, 1966, pp. 1-50.
- [5] Segall, A., "Optimal Routing for Virtual Line-Switched Data Networks", IEEE Trans. on Communications, Vol. COM-26, 1979.
- [6] Aashtiani, H.Z., "The Multi-Model Traffic Assignment Problem," Ph.D. Thesis, Sloan School of Management, M.I.T., Cambridge, Mass., May 1979.
- [7] Fratta, L., Gerla, M. and Kleinrock, L., "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design", Networks, Vol. 3, 1973, pp. 97-133.
- [8] Cantor, D.G., and Gerla, M., "Optimal Routing in a Packet Switched Computer Network", IEEE Trans. on Computers, Vol. C-23, 1974, pp.1062-1069.
- [9] Cannon, M.D. and Cullum, C.D., "A Tight Upper Bound on the Rate of Convergence of the Frank-Wolfe Algorithm", SIAM J. Control, Vol. 6, 1968, pp. 509-516.
- [10] Dunn, J.C., "Rates of Convergence of Conditional Gradient Algorithms Near Singular and Nonsingular Extremals", SIAM J. Control and Optimization Vol. 17, 1979, pp. 187-211.
- [11] Bertsekas, D.P., Gafni, E. and Vastola, K.S., "Validation of Algorithms for Optimal Routing of Flow in Networks", Proc. of IEEE Conf. on Decision and Control, San Diego, Calif., Jan. 1979, pp.220-227.
- [12] Vastola, K.S., "A Numerical Study of Two Measures of Delay for Network Routing", M.S. Thesis, Department of Electrical Engineering, Univ. of Illinois, Urbana, Ill., Sept. 1979.
- [13] Gafni, E., "Convergence of a Routing Algorithm", Lab. for Information and Decision Systems Report R-907, M.I.T., Cambridge, Mass., May 1979.
- [14] Bertsekas, D.P. and Gafni, E., "Projection Methods for Variational Inequalities with Application to the Traffic Assignment Problem", Lab. for Information and Decision Systems, Working Paper, M.I.T., Cambridge, Mass., Feb. 1980.
- [15] McQuillan, J.M., Richer, I., and Rosen E.C., "The New Routing Algorithm for the ARPANET", Bolt, Beranek and Newman, Inc., Report, May 1979.

Distribution List

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 Copies
Assistant Chief for Technology Office of Naval Research, Code 200 Arlington, Virginia 22217	1 Copy
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 Copies
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, Massachusetts 02210	1 Copy
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, Illinois 60605	1 Copy
Office of Naval Research Branch Office, Pasadena 1030 East Greet Street Pasadena, California 91106	1 Copy
Naval Research Laboratory Technical Information Division, Code 2627 Washington, D.C. 20375	6 Copies
Dr. A. L. Slafkosky Scientific Advisor Commandant of the Marine Corps (Code RD-1) Washington, D.C. 20380	1 Copy

Office of Naval Research
Code 455
Arlington, Virginia 22217
1 Copy

Office of Naval Research
Code 458
Arlington, Virginia 22217
1 Copy

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, California 92152
1 Copy

Mr. E. H. Gleissner
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084
1 Copy

Captain Grace M. Hopper
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D.C. 20350
1 Copy

Advanced Research Projects Agency
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209
1 Copy

Dr. Stuart L. Brodsky
Office of Naval Research
Code 432
Arlington, Virginia 22217
1 Copy

Captain Richard L. Martin, USN
Commanding Officer
USS Francis Marion (LPA-249)
FPO New York 09501
1 Copy