Proceedings of the 29th Conference
on Decision and Control
Honolulu, Hawaii • December 1990

Deriving Time Complexities for a Class of Distributed Gradient Projection-Based Optimal Routing Algorithms

John K. Antonio, MEMBER, IEEE
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907
(317) 494-6416

G. M. Huang[1], SENIOR MEMBER, IEEE
Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843
(409) 845-7476

Wei K. Tsai, MEMBER, IEEE
Department of Electrical Engineering
University of California, Irvine
Irvine, California 92717
(714) 725-2164

ABSTRACT

Gradient projection-based algorithms are a popular choice for solving optimal routing problems in large distributed data networks. In this paper, an upper bound is derived for the time complexity of the gradient projection-based optimal routing algorithm of Bertsekas, et al. [1]. The overall time complexity of the algorithm is given by the product of the complexity of each iteration and the complexity of the number of iterations needed to converge. It turns out that the complexity of each iteration is dominated by the time required to solve shortest path problems, and is therefore straightforward to estimate. On the other hand, estimating a meaningful bound for the number of iterations needed for convergence presents a formidable challenge, and is therefore the main focus of this paper. Classical results related to convergence rates of gradient projection-type algorithms require precise knowledge of the spectral content of the Hessian matrix. Unfortunately, such classical techniques are not readily applicable to the problem at hand since the eigenvalues of the Hessian depend on factors such as network topology and traffic demand patterns in a very non-tractable way. In this paper, an alternate analysis technique is developed which yields an upper bound for the number of iterations needed for convergence to within a small neighborhood of the optimal solution.

I. INTRODUCTION

The primary objective of virtually all routing algorithms is to select routes for those origin destination (OD) pairs which request data communication. A secondary objective is to insure that messages transmitted along the selected routes are delivered to the correct destinations. This later objective is accomplished by using standard techniques involving protocols and routing tables. In this paper we focus our attention on the former objective—the route selection problem. It is known that route selection has a substantial impact on the performance of data networks [1–9,12,13,15,16]. Roughly speaking, an optimal routing is a set of routes which yield the "best" network performance—based on some quantitative measure. The types of performance measures employed by most optimal routing formulations, estimate, in some sense, the average delay associated with sending a packet of data to a typical destination node.

An important issue to consider when implementing a routing algorithm in a large distributed data network is the question of whether the computation should be done in a centralized or distributed manner. Centralized implementations are fairly straightforward: a designated "central" node is sent data (about the state of the network) from the other network nodes; then, based on this information, the central node solves the optimal routing problem and broadcasts the solution back out to the network. One of the obvious problems with this type of scheme is the associated communication overhead (i.e., bottleneck). In contrast, certain distributed implementations can reduce this communication overhead, by requiring, for example, only nearest neighbor communication. Due to the potentially high degree of fault tolerance, fast convergence rates, low communication overhead, and other reasons, distributed implementations have received a great deal of attention in the literature over the past decade or so.

One fundamental question associated with distributed routing algorithms is that of convergence. Namely, since the order of events in distributed algorithms occur asynchronously (to one degree or another), the question of whether the algorithm will converge becomes a non-trivial one. In references [5] and [6], convergence of a class of distributed optimal routing algorithms is proven under very weak assumptions. Another pioneering work is that of Bertsekas [13], on distributed asynchronous computation of fixed points. More recently, the present authors have proven convergence of a class of distributed iterative aggregation algorithms [15], which have applications in optimal routing.

While the question of distributed asynchronous convergence has been addressed in the above cited works (and others), our goal in the present paper is to determine the time required for a class of optimal routing algorithms to converge. The time complexity of routing algorithms is an important practical as well as theoretical issue. In practice, it is imperative that the routing algorithm converge within a certain amount of time, otherwise the eventually arrived upon solution may be of little or no value. In this paper it is shown how network parameters such as maximum link utilization factors, traffic demands values, link capacity values, and others, effect the time required for convergence. In order to achieve meaningful bounds for the time complexity, a certain price was paid in that our model for computation is essentially synchronous (in terms of the order in which iterations are executed). However, the authors believe that the ground-work laid out in this paper should serve well as a guide for future work under more relaxed (i.e., asynchronous) assumptions. Our main time complexity results are for a class of path-formulated gradient projection-based algorithms.

II. FORMULATION OF THE OPTIMAL ROUTING PROBLEM

The following formulation uses the same notations and is based on the same approximating assumptions as set forth by Bertsekas and Gallager in reference [2].

A. *Delay Models*

Queuing theory is the primary methodological framework for analyzing network performance. Often times its use requires simplifying assumptions for the sake of mathematical tractability. Due to the complexity of realistic networks, it is typically impossible to obtain accurate quantitative delay predictions, however, the models used often provide valuable qualitative results and insights [2].

Perhaps the simplest queuing model is the so-called $M/M/1$ queuing system which consists of a single queuing station and a single server. It is assumed that customers (i.e., packets of data) arrive according to a Poisson process with rate F, and the probability distribution of the service rate is exponential with mean C. By applying Little's Theorem, the average delay for a packet to traverse link (i,j) is given by

$$D_{ij} = \frac{1}{C_{ij} - F_{ij}}, \tag{1}$$

where C_{ij} and F_{ij} denote the service rate and arrival rate respectively, associated with link (i,j).

Jackson's Theorem states that in a network of single server queues in which customers arrive from outside the network at each queue according to independent Poisson processes, the average number of outstanding packets in the (steady-state) system can be derived as if each queue in the network is an $M/M/1$

[1]Prof. Huang was supported in part by the Texas Advanced Research Program under Grant 4659, and by the National Science Foundation under Grant ECS-8900499

queue. So, for the purpose of measuring network performance, modeling the entire network with simple $M/M/1$ queues is justified.

Based on Jackson's Theorem and equation (1), our cost function is defined as a weighted sum of all link delays:

$$D(F) = \sum_{(i,j)\in\mathcal{L}} \frac{F_{ij}}{C_{ij} - F_{ij}}, \tag{2}$$

where links having more traffic flow are given higher relative weightings. Note that each term in the sum represents the average size of the queue associated with link (i,j). Therefore, $D(F)$ is an estimate of the total number of outstanding packets in the network. For the purposes of this paper, determining routes which minimize $D(F)$, for a given set of OD traffic demands, will constitute the notion of an optimal routing.

B. *The Optimal Routing Problem*

The following notation is needed in order to formally state the optimal routing problem.

Preliminary Notation:

W : The set of OD pairs requesting communication.

w : A generic OD pair in W.

r_w : The arrival rate (traffic demand) measured in packets/sec, for the OD pair w.

P_w : For the OD pair w, this is the set of all logical paths connecting the origin node to the destination node.

p : A generic path in P_w.

x_p : The flow rate on the logical path p.

The following constraint equations arise naturally due to conservation of flow.

Constraint Equations:

$$F_{ij} = \sum_{\substack{\text{all paths } p \\ \text{containing } (i,j)}} x_p, \tag{3}$$

$$\sum_{p\in P_w} x_p = r_w \qquad \text{for all } w \in W \tag{4}$$

and

$$x_p \geq 0 \qquad \text{for all } p \in P_w,\ w \in W. \tag{5}$$

Note that the cost function being minimized, see equation (2), can be expressed in terms of the path vector x, defined as $x = [x_p]_{\substack{p\in P_w \\ w\in W}}$. By combining constraint equation (3) with the definition of the path vector, the cost function of (2) can be written as

$$D(F) = D(x) \stackrel{\text{def}}{=} \sum_{(i,j)\in\mathcal{L}} \frac{K_{ij}x}{C_{ij} - K_{ij}x}, \tag{6}$$

where K_{ij} is a row vector with components equal to either zero or one. Specifically, the p^{th} component of K_{ij} is one if link (i,j) is on path p, otherwise, the p^{th} component of K_{ij} is zero.

The Path-Formulated Optimal Routing Problem:

$$\text{Given } r_w, \qquad \text{for each } w \in W, \tag{7}$$

$$\text{minimize } \{D(x)\}, \tag{8}$$

such that equations (4) and (5) are satisfied.

C. *Solving the Optimal Routing Problem*

It can be shown that the path-formulated optimal routing problem can be transformed into an equivalent box-constrained problem, see [15] for more details. Also, the function $D(x)$ is a differentiable convex function of the path vector x, see [5]. Therefore, the path-formulated optimal routing problem can be solved numerically by using well established techniques from nonlinear programming; our focus is on the gradient projection (GP) method. The main idea of the GP method is that after a step is made in the direction of the negative gradient, the result is orthogonally projected onto the positive orthant.

The iteration equation which results from applying the GP method to path-formulated optimal routing problem necessitates the definition of the first derivative length (FDL) of a path. The FDL of path p, denoted d_p, is defined by

$$d_p \stackrel{\text{def}}{=} \frac{\partial D(x)}{\partial x_p} = \sum_{\substack{\text{all links } (i,j) \\ \text{on path } p}} \frac{\partial D_{ij}(F_{ij})}{\partial F_{ij}}, \tag{9}$$

where

$$D_{ij}(F_{ij}) = \frac{F_{ij}}{C_{ij} - F_{ij}}. \tag{10}$$

Next, the minimum FDL (MFDL) paths, denoted as $\overline{p}_w$ for each $w \in W$, are defined by

$$d_{\overline{p}_w} = \min_{p\in P_w} \{d_p\}, \qquad \text{for all } w \in W. \tag{11}$$

We can now state the iteration equation associated with the GP algorithm.

$$x_p^{k+1} = \max\{0,\ x_p^k - \alpha^k (H_{pp}^k)^{-1}(d_p^k - d_{\overline{p}_w}^k)\},$$
$$\text{for all } w \in W,\ p \in P_w,\ p \neq \overline{p}_w, \tag{12}$$

where k is the iteration count. The term α^k denotes the step size and the term H_{pp}^k is a scaling factor which is related to the second derivative length of path p. It is easy to verify that the term $\alpha^k (H_{pp}^k)^{-1}(d_p^k - d_{\overline{p}_w}^k) \geq 0$, for all $p \in P_w$, and therefore, the above iteration equation need not be applied to those paths for which $x_p^k = 0$. Thus, we define the set of active paths, denoted $\tilde{P}_w^k$, as

$$\tilde{P}_w^k = \{p \in P_w | x_p^k > 0\}.$$

So, a more efficient version of the path-formulated GP algorithm is the following:

$$x_p^{k+1} = \max\{0,\ x_p^k - \alpha^k (H_{pp}^k)^{-1}(d_p^k - d_{\overline{p}_w}^k)\},$$
$$\text{for all } w \in W,\ p \in \tilde{P}_w^k,\ p \neq \overline{p}_w, \tag{13}$$

$$x_{\overline{p}_w}^{k+1} = r_w - \sum_{p\in\tilde{P}_w, p\neq\overline{p}_w} x_p^k, \qquad \text{for all } w \in W, \tag{14}$$

$$\tilde{P}_w^{k+1} = \tilde{P}_w^k \cup \overline{p}_w, \qquad \text{for all } w \in W. \tag{15}$$

The GP algorithm of equations (13) through (15) has been efficiently implemented by Bertsekas, et al. as a serial FORTRAN code [1]. This code uses a constant step size and the value of H_{pp}^k is an approximation of the p^{th} diagonal element of the Hessian matrix. The set of active paths for each $w \in W$ are initialized with a single (randomly selected) shortest hop path. Therefore, $|\tilde{P}_w^0| = 1$, for all $w \in W$.

III. Time Complexity of the GP Algorithm

A. *Basics*

The overall time complexity of the GP algorithm is given by the product of the complexity of each iteration and the complexity of the number of iterations required for convergence.

From equations (13) through (15), the complexity of each iteration is clearly dependent on the following three quantities: (i) the number of active paths: $|\tilde{P}_w^k|$; (ii) the number of OD pairs: $|W|$; and (iii) the number of nodes in the network: n. The dependence on $|\tilde{P}_w^k|$ is due to the fact that the flow on each (non-MFDL) active path must be updated according to equation (13). The dependence on $|W|$ comes from the fact that a MFDL path must be determined for each $w \in W$. Finally, the dependence on the number of nodes, say n, is due to the fact that solving shortest path problems (i.e., finding the MFDL paths) generically depend on the size of the graph. The complexity of each iteration (EI) is therefore denoted as $T_{EI}(k, |W|, n)$, where k represents the dependence on $|\tilde{P}_w^k|$. For notational convenience we define

$$\overline{T}_{EI}(n) = \max_k \{T_{EI}(k, |W|, n)\}. \quad (16)$$

$\overline{T}_{EI}(n)$ is fairly straightforward to estimate—the only slight difficulty comes in estimating the maximum number of active paths used in any single iteration. We note that the following is an obvious upper bound for $|\tilde{P}_w^k|$ (assuming $|\tilde{P}_w^0| = 1$)

$$|\tilde{P}_w^k| \leq k+1, \qquad \text{for all } w \in W, k \geq 0, \quad (17)$$

since at each iteration at most one new active path is added to each set $\tilde{P}_w^k$.

In contrast to the fairly straightforward task associated with estimating the complexity of each iteration (described above), our goal in this paper is to estimate the complexity associated with the number of iterations, say N_I, required for the GP algorithm to converge. Most of the classical results related to convergence rates of numerical algorithms depend on the values of the largest and smallest eigenvalues of the Hessian. For example, it is shown in [10] that by using a special step size rule, the convergence rate for the GP algorithm is bounded by

$$D(k+1) \leq \left[\frac{(B-b)}{(B+b)}\right]^2 D(k), \quad (18)$$

where $D(k)$ denotes the value of the cost function at iteration k, and B and b are, respectively, the largest and smallest eigenvalues of the Hessian. From equation (18) it is easy to see that if the difference $B-b$ is large (or $b \to 0$), then $[(B-b)/(B+b)]^2 \to 1$. On the other hand, if $b \to B$, then $[(B-b)/(B+b)]^2 \to 0$. Clearly, the smaller the value of $[(B-b)/(B+b)]^2$, the faster the convergence rate, which implies a better time complexity for the number of iterations for convergence. Unfortunately, the convergence rate of equation (18) has some practical problems when considering the application of the GP algorithm to the optimal routing problem. First of all, the assumed step size rule used to derive equation (18) is based on a type of line minimization technique which would be difficult to implement in a large distributed network—in practice a constant (or simple) step size rule is used. Second, it is difficult to determine a meaningful lower bound for b in terms of network parameters.

B. *Serial Versus Distributed Time Complexities*

Thus far we have not made a distinction between the time complexity of the GP algorithm relative to serial and distributed implementations. In a serial (single processor) implementation, the value for $T_{EI}(n)$ is the sum of the the time required to solve the shortest path problem for each $w \in W$ and the time required to update all active path flows in $\tilde{P}_w^k$. In a distributed implementation, a distributed shortest path algorithm is used (such as the distributed Bellman-Ford algorithm [2,14]) and each node i assumes the responsibility of updating all active path flows originating at node i. Of course one of the main difficulties associated with distributed algorithms (in general) is the asynchronous nature of the communication overhead.

For the purposes of this paper we shall assume that iteration $k+1$ is executed only after iteration k is completed. Under this simplifying assumption, the complexity for the number of iterations, i.e., N_I, is the same for both the serial and distributed implementations of the GP algorithm. In terms of the distributed implementation, this assumption implicitly assumes the existence of a uniform upper bound for the communication time complexity of each iteration. This type of assumption results in what is typically called a partially asynchronous distributed algorithm. We note that Bertsekas, et al. have proven that the distributed GP algorithm will actually converge (eventually) in a virtually totally asynchronous computing environment [5,6,13]. However, with such mild restrictions on the ordering of events it becomes much more difficult to bound the number of iterations required for convergence.

The analysis techniques introduced in the present paper for bounding the number of iterations required (for the partially asynchronous case) serve a twofold purpose. First, the resulting bound for the number of iterations may serve as an optimistic estimate of the number of iterations assuming a more asynchronous model. Second, the analysis techniques developed in deriving the bound for the partially asynchronous case may serve as a model for deriving similar results under more relaxed assumptions.

IV. The Main Results

In this section we present upper bound results for the number of iterations needed by the GP-based algorithm to converge to within a small neighborhood of the optimal solution. Three separate bounds are derived, each of which is based on one of three sets of assumptions. Each set of assumptions consists of four assumptions; three of which are common to all sets.

Next, we list all necessary notation for stating the main results.

A. *Notation*

W : The set of OD pairs requesting communication.

w : A generic OD pair in W.

r_w : The arrival rate for the OD pair $w \in W$.

$r_{\min}$: The minimum arrival rate, for all $w \in W$:

$$r_{\min} = \min_{w \in W}\{r_w\}.$$

P_w : For the OD pair w, this is the set of logical paths which connect the origin node to the destination node.

p : A generic path in P_w.

k : The iteration count.

x_p^k : The flow rate on path p at iteration k.

$\tilde{P}_w^k$: This is the set of active paths in P_w at iteration k:

$$\tilde{P}_w^k = \{p \in P_w | x_p^k > 0\}.$$

F_{ij}^k : The flow rate on link (i,j) at iteration k.

C_{ij} : The capacity of link (i,j).

$C_{\min}$: The minimum value of C_{ij}, for all $(i,j) \in \mathcal{L}$.

$C_{\max}$: The maximum value of C_{ij}, for all $(i,j) \in \mathcal{L}$.

ρ_{ij}^k : The utilization factor of link (i,j) at iteration k:

$$\rho_{ij}^k = \frac{F_{ij}^k}{C_{ij}}.$$

$\rho_{\max}$: The maximum link utilization factor for all $(i,j) \in \mathcal{L}$, and all k:

$$\rho_{\max} = \max_{\substack{(i,j)\in\mathcal{L} \\ k=1,2,\ldots}} \left\{\rho_{ij}^k\right\}.$$

$D(k)$: The value of the cost function at iteration k:

$$D(k) = \sum_{(i,j)\in\mathcal{L}} \frac{F_{ij}^k}{C_{ij} - F_{ij}^k} = \sum_{(i,j)\in\mathcal{L}} \frac{K_{ij}x^k}{C_{ij} - K_{ij}x^k}.$$

D^* : The optimal value of the cost function.

$e(k)$: The normalized error:

$$e(k) = \frac{D(k) - D^*}{D^*}.$$

h_w : The minimum hop distance between the origin and destination of each OD pair $w \in W$.

$h_{\max}$: The maximum value of h_w, for all $w \in W$:

$$h_{\max} = \max_{w \in W}\{h_w\}.$$

$h_{\min}$: The minimum value of h_w, for all $w \in W$:

$$h_{\min} = \min_{w \in W}\{h_w\}.$$

h_{avg} : The average value of h_w, for all $w \in W$:

$$h_{\text{avg}} = \frac{1}{|W|}\sum_{w \in W} h_w.$$

B. Assumptions

(A1) Iteration k is completed before iteration $k+1$ begins.

(A2) $0 \leq \rho_{\max} < 1$.

(A3) There exists constants $\zeta_{\min}$ and $\zeta_{\max}$, such that α^k (the stepsize at iteration k) satisfies

$$(\zeta_{\min})\overline{\alpha}^k \leq \alpha^k \leq (\zeta_{\max})\overline{\alpha}^k,$$

where

$$0 < \zeta_{\min} \leq \zeta_{\max} < 1$$

and

$$\overline{\alpha}^k \stackrel{\text{def}}{=} \left(\frac{C_{\min}}{C_{\max}}\right)^3 \left(\frac{(1-\rho_{\max})^5}{3}\right) \left(\frac{\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}}{\max_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}}\right) \left(\frac{(h_{\min})}{(|W|)(h_{\text{avg}})}\right).$$

(A4.1) There exists a constant K_1 such that

$$\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\} \geq (K_1)(k+1), \qquad \text{for all } k.$$

(A4.2) There exists a constant K_2 such that

$$\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\} \geq (K_2)\sqrt{k+1}, \qquad \text{for all } k.$$

(A4.3) There exists a constant K_3 such that

$$\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\} \geq K_3, \qquad \text{for all } k.$$

The need for assumption (A1) was mentioned previously. Namely, without assuming this type of synchronization (with respect to the execution of the iterations) the analysis of convergence rates becomes much more difficult. (Note: In a more relaxed asynchronous computing environment the question of interest is not typically one of "How fast does the algorithm converge?" but more fundamentally "Will the algorithm converge?") Assumption (A2) ensures that a valid routing solution exists, i.e., $F_{ij}^k < C_{ij}$, for all $(i,j) \in \mathcal{L}, k$. Assumption (A3) requires the stepsize to lie within a specified interval. (Note: It is well-known—for general gradient-type algorithms—that if the stepsize is too large, then the algorithm may not converge. On the other hand, if the stepsize is too small, then the convergence rate may be arbitrarily slow.) Assumptions (A4.1) through (A4.3) are three different lower bound assumptions for the size of $\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}$. Actually, (A4.3) is always satisfied since $\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\} \geq 1 (= K_3)$. Assumption (A4.1) states that $\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}$ increases linearly with the number of iterations, while (A4.2) assumes $\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}$ increases with the square root of k.[2]

C. Upper Bounding the Number of Iterations for Convergence

Due to space limitations, only a sketch of the proof for Theorem 1 is given in the Appendix. Detailed proofs are included in [18].

[2]Actually, similar complexity results as those presented in this paper can be derived under the assumption that $\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}$ is bounded below by an arbitrary function of k, say $f(k)$, which satisfies $1 \leq f(k) \leq k+1$.

Theorem 1: Given that assumptions (A1), (A2), (A3), and (A4.1) are satisfied, then for any $\epsilon > 0$, the number of iterations required for the GP-algorithm to converge to $e(N_I) \leq \epsilon$ is bounded by

$$N_I \leq B_1 \stackrel{\text{def}}{=} \left(\frac{\sqrt{24}(\zeta_{\max})^{0.5}(\rho_{\max})^{0.5}(C_{\max})^{4.5}(1+C_{\max})}{\epsilon(\zeta_{\min})(1-\zeta_{\max})^{0.5}(r_{\min})(C_{\min})^{4.5}(K_1)}\right) \left(\frac{1}{(1-\rho_{\max})^8}\right)\left(\frac{h_{\max}}{h_{\min}}\right).$$

Theorem 2: Given that assumptions (A1), (A2), (A3), and (A4.2) are satisfied, then for any $\epsilon > 0$, the number of iterations required for the GP-algorithm to converge to $e(N_I) \leq \epsilon$ is bounded by

$$N_I \leq B_2 \stackrel{\text{def}}{=} \left(\frac{24(\zeta_{\max})(\rho_{\max})(C_{\max})^9(1+C_{\max})^2}{\epsilon^2(\zeta_{\min})^2(1-\zeta_{\max})(r_{\min})^2(C_{\min})^9(K_2)^2}\right) \left(\frac{1}{(1-\rho_{\max})^{16}}\right)\left(\frac{h_{\max}}{h_{\min}}\right)^2.$$

Theorem 3: Given that assumptions (A1), (A2), (A3), and (A4.3) are satisfied, then for any $\epsilon > 0$, the number of iterations required for the GP-algorithm to converge to $e(N_I) \leq \epsilon$ is bounded by

$$N_I \leq B_3 \stackrel{\text{def}}{=} \exp\left\{\left(\frac{24(\zeta_{\max})(\rho_{\max})(C_{\max})^9(1+C_{\max})^2}{\epsilon^2(\zeta_{\min})^2(1-\zeta_{\max})(r_{\min})^2(C_{\min})^9(K_3)^2}\right) \left(\frac{1}{(1-\rho_{\max})^{16}}\right)\left(\frac{h_{\max}}{h_{\min}}\right)^2\right\}.$$

D. Discussion of the Results

From the above three bounds, note that for any fixed set of parameters (i.e., $\rho_{\max}$, $\zeta_{\max}$, ϵ, etc.) we have $B_1 < B_2 < B_3$ (since $B_2 = (B_1)^2$, $B_3 = \exp\{B_2\}$ and $B_1 \geq 1$). Therefore, our bounds indicate that the number of iterations for convergence decreases as the function lower bounding $\min_{w \in W}\left\{\left|\tilde{P}_w^k\right|\right\}$ is increased.

We now make some observations which apply to all three bounds. Assuming that all other parameters are fixed, then:

(i) as $\rho_{\max} \to 1$, the bounds increase,

(ii) as $\epsilon \to 0$, the bounds increase,

(iii) as $\zeta_{\min} \to 0$, the bounds increase,

(iv) as $\zeta_{\max} \to 1$, the bounds increase,

(v) the bounds are proportional to $\frac{C_{\max}}{C_{\min}}$,

(vi) the bounds are proportional to $\frac{h_{\max}}{h_{\min}}$.

Observation (i) agrees with our experience with the GP algorithm. In particular, we have found through experimentation that by increasing the total demand (so as to increase $\rho_{\max}$) the number of iterations required for convergence can be made arbitrarily large, as $\rho_{\max} \to 1$. Observation (ii) coincides with the generally accepted fact that the convergence rate of gradient-based algorithms tend to decrease near the optimal solution. Observations (iii) and (iv) confirm the fact that if the stepsize is either too large or too small, then the rate of convergence is decreased (and thus the number of iterations for convergence is increased). Observations (v) and (vi) give us new insights. Observation (v) indicates that, if possible, it may be beneficial to have $C_{\min} = C_{\max}$, in order to decrease the number of iterations for convergence. Observation (vi) is perhaps the most non-intuitive and interesting aspect of our results. Note that depending on the relationship of the set of active OD pairs to the topology of the network graph, the value of $\frac{h_{\max}}{h_{\min}}$ may or may not depend on the size of the network, i.e., the number of nodes. In any event, we note that the value of $\frac{h_{\max}}{h_{\min}}$ is bounded above by the diameter of the network graph, which in many practi-

cal situations is bounded by a slowly increasing function of the number of nodes.

V. Conclusions and Extensions

We have presented bounds for the number of iterations required for a class of path-formulated gradient projection-based algorithms to converge. The bounds confirm observations made through experimentation and experience, and more importantly also offer new insights. Our results show that the number of iterations required for convergence increases rapidly as the maximum link utilization factor increases. Also, our results show that the number of iterations for convergence is dependent on the ratio of the maximum {minimum-hop-distance} to the minimum {minimum-hop-distance}, taken over all communicating origin-destination pairs. Thus, in the worst case, the number of iterations may be dependent on the diameter of the network, however, it is possible that the number of iterations is independent of the problem size (given that the aforementioned ratio is a constant).

We plan two directions of future work. First, we shall attempt a similar derivation of bounds for a competing link formulated optimal routing algorithm [17]. Second, we will try to extend the results of the present paper to include the class of iterative aggregation/disaggregation algorithms described in [15]. It seems possible to show that by aggregating paths associated with appropriately chosen groups of OD pairs, the number of iterations for convergence may decrease. By appropriately chosen groups of OD pairs, we mean groups in which any of the OD pairs in a particular group is separated by approximately the same number of hops.

Appendix

The purpose of this appendix is to give an indication of the main ideas involved in proving Theorem 1. (We should note that the proofs of Theorems 2 and 3 are very similar to the proof of Theorem 1.) The actual detailed proof, which is included in reference [18], involves stating and proving a string of nine preliminary lemmas. Lemma 9 (which is stated below without proof) is a lower bound result for the amount of decrease in the normalized error from iteration k to iteration $k+1$, and is the cornerstone of proving Theorems 1, 2, and 3. In addition to Lemma 9, Lemma 4 is also stated below without proof. Assumptions (A1) through (A3) are assumed for Lemmas 4 and 9.

Lemma 4:

$$\text{(i)} \quad D^* \geq \left(\frac{r_{\min}}{C_{\max}}\right)(|W|)(h_{\text{avg}}),$$

$$\text{(ii)} \quad D(0) \leq \left(\frac{\rho_{\max}}{1-\rho_{\max}}\right)(|W|)(h_{\text{avg}}).$$

Lemma 9: For any $\epsilon > 0$, if $e(k) > \epsilon$, then the decrease in the normalized error from iteration k to iteration $k+1$ is bounded below by

$$e(k) - e(k+1) \geq \left(\frac{(1-\zeta_{\max})(r_{\min})(C_{\min})^9(1-\rho_{\max})^{15}}{6(\zeta_{\max})(C_{\max})^8(1+C_{\max})^2}\right) \epsilon^2(\zeta_{\min})^2 \left(\frac{\left(\min_{w\in W}\left\{\left|\tilde{P}_w^k\right|\right\}\right)^2 (h_{\min})^2}{\left(\max_{w\in W}\left\{\left|\tilde{P}_w^k\right|\right\}\right)(h_{\max})^2}\right).$$

Theorem 1: Given that assumptions (A1), (A2), (A3), and (A4.1) are satisfied, then for any $\epsilon > 0$, the number of iterations required for the GP-algorithm to converge to $e(N_I) \leq \epsilon$ is bounded by

$$N_I \leq \left(\frac{\sqrt{24}(\zeta_{\max})^{0.5}(\rho_{\max})^{0.5}(C_{\max})^{4.5}(1+C_{\max})}{\epsilon(\zeta_{\min})(1-\zeta_{\max})^{0.5}(r_{\min})(C_{\min})^{4.5}(1-\rho_{\max})^8(K_1)}\right)\left(\frac{h_{\max}}{h_{\min}}\right).$$

Proof of Theorem 1: Note from Lemma 9 that for all $k = 0, 1, 2, \ldots$, the normalized error decreases from iteration k to $k+1$. Therefore, $\max_k\{e(k)\} = e(0)$. By applying assumption (A4.1) to Lemma 9 and letting $\max_{w\in W}\left\{\left|\tilde{P}_w^k\right|\right\} = k+1$, we have that

$$\begin{aligned} e(0) - e(N_I) &\geq \left(\frac{(1-\zeta_{\max})(r_{\min})(C_{\min})^9(1-\rho_{\max})^{15}}{12(\zeta_{\max})(C_{\max})^8(1+C_{\max})^2(h_{\max})^2}\right) \epsilon^2(\zeta_{\min})^2(h_{\min})^2(K_1)^2(1+2+3+\cdots N_I) \\ &\geq \left(\frac{(1-\zeta_{\max})(r_{\min})(C_{\min})^9(1-\rho_{\max})^{15}}{12(\zeta_{\max})(C_{\max})^8(1+C_{\max})^2(h_{\max})^2}\right) \epsilon^2(\zeta_{\min})^2\left(\frac{(h_{\min})^2(K_1)^2}{2}\right)(N_I)^2 \quad (A.8) \end{aligned}$$

Next, note that we can bound $e(0)$ as follows:

$$\begin{aligned} e(0) &= \frac{D(0) - D^*}{D^*} \\ &\leq \frac{D(0)}{D^*} \\ &\leq \left(\frac{\rho_{\max}}{1-\rho_{\max}}\right)\frac{(|W|)(h_{\text{avg}})}{D^*} \\ &\quad \text{(By Lemma 4 – part (ii))} \\ &\leq \left(\frac{\rho_{\max}}{1-\rho_{\max}}\right)\left(\frac{C_{\max}}{r_{\min}}\right) \qquad (A.9) \\ &\quad \text{(By Lemma 4 – part (i))} \end{aligned}$$

Now, by setting $e(N_I) = 0$ and applying equation $(A.9)$ to equation $(A.8)$, we get

$$(N_I)^2 \leq \left(\frac{\rho_{\max}}{1-\rho_{\max}}\right)\left(\frac{24(\zeta_{\max})(C_{\max})^9(1+C_{\max})^2}{\epsilon^2(\zeta_{\min})^2(1-\zeta_{\max})(r_{\min})^2(C_{\min})^9}\right)\left(\frac{(h_{\max})^2}{(1-\rho_{\max})^{15}(K_1)^2(h_{\min})^2}\right)$$

which implies that

$$N_I \leq \left(\frac{\sqrt{24}(\zeta_{\max})^{0.5}(\rho_{\max})^{0.5}(C_{\max})^{4.5}(1+C_{\max})}{\epsilon(\zeta_{\min})(1-\zeta_{\max})^{0.5}(r_{\min})(C_{\min})^{4.5}(1-\rho_{\max})^8(K_1)}\right)\left(\frac{h_{\max}}{h_{\min}}\right).$$

□

References

[1] D. P. Bertsekas, B. Gendron and and W. K. Tsai, "Implementation of an optimal multicommodity network flow algorithm based on gradient projection and a path flow formulation," Mass. Inst. Technol., Cambridge, MA, Tech. Rep. LIDS-P-1364, Feb., 1984.

[2] D. P. Bertsekas and R. G. Gallager, *Data Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[3] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks.* Princeton, NJ: Princeton University Press, 1962.

[4] M. Schwartz, "Analysis of congestion control techniques in computer routing and flow control in data networks," *NATO Advanced Study Inst.: New Concepts in Multi-User Communications*, Norwich, U.K., Aug. 4–16, 1980; Sijthoof and Nordhoof, Neth.

[5] J. N. Tsitsiklis and D. P. Bertsekas, "Distributed asynchronous optimal routing for data networks," *IEEE Trans. Automat. Contr.* vol. AC-31, pp. 325–332, 1986.

[6] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Automat. Contrl.*, vol. AC-31, pp. 803–812, Sept. 1986.

[7] F. Kamoun, and L. Kleinrock, "Stochastic performance evaluation of hierarchical routing for large networks," *Comput. Networks*, vol. 3, pp. 337–353, 1979.

[8] W. K. Tsai, "Convergence of gradient projection routing methods in an asynchronous stochastic quasi-static virtual circuit network," *IEEE Transactions on Automat. Contr.*, vol. 34, pp. 20–33, Jan. 1989.

[9] M. Schwartz and T. E. Stern, "Routing techniques used in computer communication networks," *IEEE Trans. Commun.*, vol. COM-28, pp. 539–552, Apr. 1980.

[10] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming.* Reading, MA: Addison-Wesley, 1973.

[11] R. S. Varga, *Matrix Iterative Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1962.

[12] W. K. Tsai, "Optimal quasi-static routing for virtual circuit networks subjected to stochastic inputs," Ph.D. Dissertation, Dep. EECS, Mass. Inst. Technol., Cambridge, MA, 1986.

[13] D. Bertsekas, "Distributed asynchronous computation of fixed points," *Math. Programm.*, vol. 27, pp. 107–120, 1983.

[14] A. V. Aho, J. E. Hoptcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Reading, MA: Addison-Wesley, 1974.

[15] W. K. Tsai, G. M. Huang, J. K. Antonio, and W. T. Tsai, "Distributed iterative aggregation algorithms for box-constrained minimization problems and optimal routing in data networks," *IEEE Trans. on Auto. Control,* Vol. 34, pp. 34–46, Jan. 1989.

[16] J. K. Antonio, "Fast distributed and parallel algorithms for data network control problems," Ph.D. Dissertation, Dep. of Electrical Engineering, Texas A&M University, College Station, TX, 1989.

[17] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. on Communications,* Vol. 25, pp. 73–85, Jan. 1977.

[18] J. K. Antonio, W. K. Tsai, and G. M. Huang, "Time complexity of a path-formulated gradient projection-based optimal routing algorithm," submitted to *IEEE Trans. on Auto. Control,* on July 11, 1990.