Proceedings of the 32nd Conference
on Decision and Control
San Antonio, Texas · December 1993

Computational Experience of Implementing a Distributed Asynchronous Algorithm with Stochastic Delays in Routing Networks

Bassem F. Beidas and George P. Papavassilopoulos
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-2563

Abstract

A distributed asynchronous algorithm that solves an optimal routing for a network that connects various U.S. cities is considered. The communication delays among the processors are assumed to be stochastic with Markovian character.

Results of the extensive simulation that we implemented assert the practical applicability of distributed asynchronous algorithms with stochastic delays. Comparison results for varying the probability distribution of these delays are provided. The impact of varying the communication delay bound and the stepsize is also assessed.

1 Introduction

The recent emphasis on parallel processing is motivated by the compelling need to accelerate computations when solving large dimensional problems in which great memory storage and immense computation capabilities may hinder the performance of centralized algorithms. A number of processors are utilized that operate simultaneously in a collaborative manner on several subproblems decomposed from the original one. To further amend the enhancement of performance, the processors are permitted to communicate asynchronously such that little coordination of communication is maintained. It is shown that dispensing with the synchronization points at the end of each iteration induces improved efficiency, load balancing among processors and reduction of processor idle periods [4, 6, 8, 16].

Tsitsiklis, Bertsekas and Athans [15] proposed asynchronous implementation for solving optimization problems which seems to offer the initial work of the current investigation. Since then, several algorithms suitable for operation in a multiprocessor environment emerged such that diverse areas are covered. Kushner and Yin [9] studied stochastic approximation techniques for parallel processing using the ODE approach. In their analysis, the approach of weak convergence was utilized. Kaszkurewicz, Bhaya and Šiljak [10] implemented asynchronous iterations to solve a class of nonlinear problems and derived results that retained the quasi-dominance conditions previously studied by Šiljak [12] for the synchronous case. Furthermore, Üresin and Dubois [16] proposed asynchronous algorithms that deal with nonnumerical methods such as symbolic computation and artificial intelligence applications.

In this paper, we provide a computational experience of utilizing distributed asynchronous algorithms with stochastic delays that solve minimization problems over a constraint set. The plausibility of the notion of stochastic delays stems from the fact that it models the case of unpredictable delays in the communication among the processors and therefore addresses various reliability aspects [3]. Constrained optimization problems are prevalent in actual applications, where the nature of the problems solved necessitates imposing natural conditions. Other cases are when the designer often wishes to confine the acceptable values of processors' iterates to lie within a certain region in order to further prevent the processors from straying away from the correct solution. Our main interest in this paper is to measure performance and estimate efficiency of the distributed asynchronous algorithms with stochastic delays. We also obtain comparison results of distributed asynchronous algorithms with stochastic delays and their deterministic delays counterpart algorithms of the same problem under duplicate conditions. In particular, we apply these algorithms to treat optimal routing of data communication that best exemplifies nonlinear multicommodity network flow problems. It is often the norm to find that the size of these problems is overwhelmingly large that a set of processors operating in distributed fashion is required to provide the anticipated solution. See [2] for further results on distributed asynchronous algorithms with stochastic delays.

The paper is organized as follows. In Section 2 we devise the distributed asynchronous algorithm that solves optimal routing of data communication networks. A test problem of a network that routes data among several interconnected U. S. cities and its computational results are given in Section 3. Finally, we discuss conclusions in Section 4.

2 Routing Network

Suppose that we are given a network of nodes and arcs and a set W of ordered pairs w of distinct nodes referred to as the origin-destination (OD) of w. We are also given that r_w (measured in data units/seconds) is the arrival rate of traffic entering the network at the origin and exiting at the destination of w. We denote P_w as the set of all simple paths that connect the origin and destination of w, x_p as the flow routed through path p and x_w as all the path flows $x_p \in P_w$. The fundamental constraints imposed on this problem require the conservation of the load when shared among the various paths and maintaining the nonnegativity property of the path flows which yield

$$\sum_{p \in P_w} x_p = r_w, \forall w \in W, \tag{1}$$

$$x_p \geq 0, \forall p \in P_w, \forall w \in W. \tag{2}$$

Define F_{ij} as the total flow of arc (i, j)

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

Consider a cost function such that

$$D(x) = \sum_{(i,j)} D_{ij}(F_{ij}). \tag{4}$$

Our objective is to find the set of paths for each origin-destination pair and the amount of flow routed along each path such that the cost function defined in equation (4) is minimized.

The distributed asynchronous algorithm of this section is carried out along the lines of the one studied by Tsitsiklis and Bertsekas [14]. It takes the gradient projection form in which processor w is responsible for updating x_w. At every step the update is in the direction opposite to the gradient of the cost function and whenever a processor detects that its iteration is excluded from the constraint set, it enforces feasibility by projecting its iteration back onto the feasible set.

In a manner reminiscent of the ARPANET algorithm [11], the end node of any arc ascertains the amount of flow through that arc by averaging of some previous values of the total flow of the arcs F_{ij}. We therefore write

$$\bar{F}_{ij}(t) = \sum_{t'=t-T}^{t} c_{ij}(t,t')F_{ij}(t'), \qquad (5)$$

where $c_{ij}(t,t')$ are nonnegative scalars such that

$$\sum_{t'=t-T}^{t} c_{ij}(t,t') = 1$$

and T is the bound over which the averaging is implemented. The averaged value of the total flow of arcs $\bar{F}_{ij}$ is then propagated to all other nodes. Due to the distributed asynchronous nature of the algorithm, this information may be received with a delay. Consequently, processor w computes

$$\lambda_p(t) = \sum_{(i,j)\in p} D'_{ij}(\bar{F}_{ij}(t+1-d^w(t))), \qquad (6)$$

where $'$ denotes the derivative and λ_w is all the λ_p for the paths $p \in P_w$. We note that $d^w(t)$ is the communication delay encountered when sending information to processor w at time t. The next step casts the equality constraints in a form that is forthcoming to parallelization by means of transforming the equality constraints to nonnegativity constraints. Then each processor would be able to project onto the positive orthant at its own pace and independently of the other processors. Consequently, processor w finds the minimum first derivative length (MFDL) [5] path $\bar{p}_w(t)$ such that

$$\lambda_{\bar{p}_w(t)}(t) = \min_{p\in P_w} \lambda_p(t) \qquad (7)$$

In practice, there exist transients in the total flows F^{ij} that occur as the routing changes. This contributes to having a settling time quite substantial to be ignored that renders processor w incapable of computing actual flows. Processor w will in turn use $\lambda_w(t)$ to compute desired path flows $\bar{x}_w(t)$, whose components are $\bar{x}_p(t)$ for $p \in P_w$. Therefore, the actual values do not assume their desired values instantaneously. Instead, the actual flow $x_p(t)$ takes some value between $\bar{x}_p(t)$ and $x_p(t-1)$. We assume that there exist scalars $\alpha > 0$, $a_p(t)$, such that

$$a_p(t) \geq \alpha, \ \forall p, t, \qquad (8)$$

and

$$x_p(t+1) = a_p(t)\bar{x}_p(t) + (1-a_p(t))x_p(t), \ \forall p, t. \qquad (9)$$

The distributed asynchronous algorithm is described as follows. For any $p \in P_w$, $p \neq \bar{p}_w(t)$

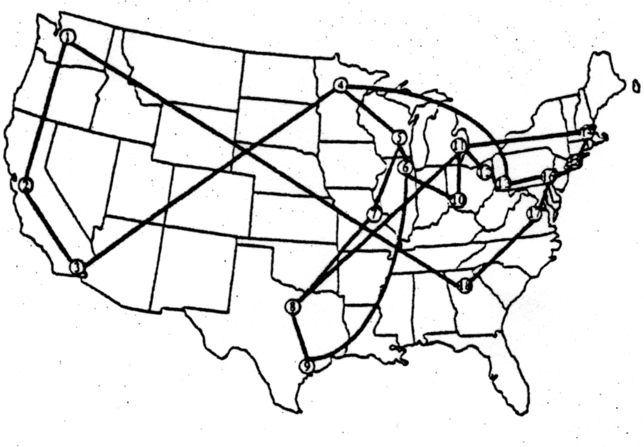
$$\bar{x}_p(t) = \max\left\{0, x_p(t) - \frac{\gamma}{H_p(t)}(\lambda_p(t) - \lambda_{\bar{p}_w(t)}(t))\right\}, \qquad (10)$$

where H_p is the second derivative length

$$H_p(t) = \sum_{(i,j)\in L_p} D''_{ij}(F_{ij}(t)) \qquad (11)$$

and L_p is the set of arcs belonging to either p or the corresponding minimum first derivative path $\bar{p}_w$, but not both. For $p = \bar{p}_w(t)$, we have

$$\bar{x}_{\bar{p}_w(t)}(t) = r_w - \sum_{p\in P_w,\ p\neq \bar{p}_w(t)} \bar{x}_p(t) \qquad (12)$$



1. Seattle	10. Cincinatti
2. San Francisco	11. Detroit
3. Los Angeles	12. Cleveland
4. Minneapolis	13. Pittsburg
5. Milwaukee	14. Boston
6. Chicago	15. New York
7. St. Louis	16. Philadelphia
8. Dallas	17. Washington
9. Houston	18. Atlanta

Figure 1: The network topology.

3 Simulation Studies

In this section we use computer simulation to measure performance and estimate efficiency of distributed asynchronous algorithms with stochastic delays in the context of data communication optimal routing networks.

In our test problem, we considered the network topology depicted in Figure 1 where all the connections are assumed to carry the flows bidirectionally. This network connects various U. S. cities with the intention of routing data as follows.

r_1 : load to be routed from Seattle to Detroit
r_2 : load to be routed from Detroit to Seattle
r_3 : load to be routed from Chicago to Washington
r_4 : load to be routed from Washington to Chicago
r_5 : load to be routed from Houston to Atlanta
r_6 : load to be routed from Atlanta to Houston.

In all the simulations, we considered the cost function at every arc to be quadratic

$$D_{ij} = \frac{1}{2}(F_{ij})^2, \text{ for every } (i,j), \qquad (13)$$

the traffic loads r_w for all w were equal to 3.0 and the parameter T in (5) was chosen to be 4. In addition, the parameter α in (8) was chosen to be 0.25. The distributed asynchronous algorithms with stochastic delays were tested on different initial conditions to show the uniqueness of the minimum point.

In our simulation, the delays $d^w(t)$ assume their values from the set $\{1, 2, \cdots, B\}$, where B is the communication delay bound and the delays were either sequences of Markov chains or independently generated with probabilities equal to the limiting behavior of these Markov delays. The delays were generated according to several probability distributions and the initial delays were chosen at random with equal probabilities. The probability distributions of these delays are relegated in the Appendix.

It is assumed that the algorithm terminates at time t if the termination function TF(t) meets the tolerance level of 0.001, i.e.,

$$TF(t) = \max_{\tau,\tau' \in \{t-B+1,\cdots,t\}} E\|x(\tau) - x(\tau')\| \leq 0.001 \qquad (14)$$

We computed $E\| \cdot \|$ by averaging over 100 trials of the experiment. For each trial a different computer realization of the delays was used. Since the previous values of the iterations that lie within the delay bound affect the value at which the algorithm stands, the above termination criterion is necessary to ensure that all of these values have also been stabilized.

We started by simulating the behavior of distributed asynchronous algorithms with bounded delays. In essence, the delays were independently generated from a uniform distribution on the set $\{1, 2, \cdots, B\}$. Figure 2 plots the performance of these algorithms as measured by the termination time in terms of varying the stepsize for the delay bounds $B = 4$, 8 and 12.

Next we tested the case of ordered scheduling that is discussed in Beidas and Papavassilopoulos [3]. A reasonable restriction is imposed where we assume that the information is received in the order it was produced.

We assume that each processor w has a local memory where the latest x_w generated at time instant t is kept and when the new information arrives it is labelled using a time stamp as to when it was computed by the other processors. If it happens that this processor acknowledges that the information it receives was generated at a time instant earlier than t, then processor w will discard it. Therefore, the probability distribution enforces that

$$\Pr\{d^w(t) > d^w(t-1) + 1\} = 0, \text{ for all } w \text{ and } t, \qquad (15)$$

which entails that the entries of the probability matrices for the delays that are above the superdiagonal to be zeros.

We simulated the distributed asynchronous algorithms with Markov delays having the property defined by equation (15) and the distributed asynchronous algorithms with independent delays whose probabilities are equal to the limiting behavior of these Markov delays. Figure 3 and Figure 4 illustrate the performance of these algorithms for different probability distributions and for $B = 4$ and $B = 8$, respectively.

In general, we make the following observations. Firstly, for small stepsize γ, changing the probability distribution has little discernible effect on the termination time. Secondly, the performance curves of the algorithms with Markov delays follow those of the algorithms with independent delays, corroborating our intuition. Thirdly, the performance curve which corresponds to the probability distribution $PD1$ that places emphasis on recent values of the delays spans more values of stepsizes before the eventual divergence.

From Figure 3 and Figure 4, we notice that the performance curves of the Markov delays lag those of the independent delays. This is attributed to the fact that the time needed to bring the Markov delays to their limiting behavior causes the algorithms with Markov delays to converge more slowly than the algorithms with independent delays. This effect is more prevailing for the case of $B = 8$ where for some stepsizes the algorithms with independent delays converge while their Markov delay counterparts do not.

To underscore the above phenomenon, we obtained Figure 5 and Figure 6 by plotting one run of the performance of algorithms with Markov delays for $\gamma = 1.095$, probability distribution $PD1$ and $\gamma = 0.845$, probability distribution $PD2$. The same was done for the algorithms with independent delays. From both figures we notice that while the independent delays case converges to the minimum value of the cost function, the Markov delays case exhibits severe oscillatory behavior around the minimum.

Next, we examined the effects of the stepsize γ and the communication delay bound B on distributed asynchronous algorithms with Markov delays. Figure 7 depicts the termination time as the delay bound B and stepsize γ are varied. Noteworthy of mention is that Figure 3 and Figure 4 show that algorithms with stepsize γ chosen to be 0.05, 0.1 and 0.2 display constant termination time as the probability distribution is changed. We note that the termination time grows quickly with decreasing stepsize γ. In addition, it grows faster with increasing delay bound B when stepsize γ is small than it does when γ is large.

4 Conclusion

With the aid of simulation studies we estimated the performance of distributed asynchronous iterations with stochastic delays and obtained comparison results as the probability distribution of these delays changed. We also assessed the impact of varying the communication delay bound and the stepsize on the termination of these algorithms and it was shown that the performance of distributed asynchronous algorithms is in fact predictable.

Appendix

We provide the probability distribution for the delays that result from the distributed asynchronous nature of the algorithms when solving the optimal routing network of U.S. cities given in Section 3.

First, when the communication delay bound $B = 4$, the probability matrices that characterize probability distribution $PD1$ are given below.

$$P^1 = P^4 = \begin{bmatrix} .75 & .25 & 0 & 0 \\ .65 & .2 & .15 & 0 \\ .6 & .2 & .1 & .1 \\ .25 & .25 & .25 & .25 \end{bmatrix}.$$

$$P^2 = P^5 = \begin{bmatrix} .6 & .4 & 0 & 0 \\ .5 & .2 & .3 & 0 \\ .25 & .25 & .25 & .25 \\ .625 & .125 & .125 & .125 \end{bmatrix}$$

$$P^3 = P^6 = \begin{bmatrix} .8 & .2 & 0 & 0 \\ .4 & .3 & .3 & 0 \\ .5 & .2 & .2 & .1 \\ .1 & .34 & .25 & .31 \end{bmatrix}$$

Second, the probability matrices that characterize probability distribution $PD2$ are given below.

$$(P^i,\ i = 1, \cdots, 6) = \begin{bmatrix} .5 & .5 & 0 & 0 \\ .4 & .3 & .3 & 0 \\ .25 & .25 & .25 & .25 \\ .25 & .25 & .25 & .25 \end{bmatrix}$$

Third, the probability matrices that characterize probability distribution $PD3$ are given below.

$$P^1 = P^4 = \begin{bmatrix} .2 & .8 & 0 & 0 \\ .2 & .25 & .55 & 0 \\ .1 & .2 & .1 & .6 \\ .25 & .25 & .25 & .25 \end{bmatrix}$$

$$P^2 = P^5 = \begin{bmatrix} .375 & .625 & 0 & 0 \\ .3 & .2 & .5 & 0 \\ .25 & .25 & .25 & .25 \\ .125 & .125 & .125 & .625 \end{bmatrix}$$

$$P^3 = P^6 = \begin{bmatrix} .5 & .5 & 0 & 0 \\ .1 & .3 & .6 & 0 \\ .1 & .2 & .2 & .5 \\ .1 & .34 & .25 & .31 \end{bmatrix}$$

Now we provide the probability distributions when the communication delay bound $B = 8$. First, the probability matrices that characterize probability distribution $PD1$ are given below.

$$P^1 = P^4 = \begin{bmatrix} .75 & .25 & 0 & 0 & 0 & 0 & 0 & 0 \\ .55 & .1 & .35 & 0 & 0 & 0 & 0 & 0 \\ .25 & .25 & .25 & .25 & 0 & 0 & 0 & 0 \\ .33 & .33 & .21 & .06 & .07 & 0 & 0 & 0 \\ .1 & .15 & .25 & .25 & .2 & .05 & 0 & 0 \\ .3 & .15 & .15 & .1 & .1 & .1 & .1 & 0 \\ .125 & .125 & .125 & .125 & .125 & .125 & .125 & .125 \\ .4 & 0 & 0 & .2 & .1 & .1 & .1 & .1 \end{bmatrix}$$

$$P^2 = P^5 = \begin{bmatrix} .5 & .5 & 0 & 0 & 0 & 0 & 0 & 0 \\ .45 & .3 & .25 & 0 & 0 & 0 & 0 & 0 \\ .25 & .25 & .25 & .25 & 0 & 0 & 0 & 0 \\ .625 & .125 & .125 & .125 & 0 & 0 & 0 & 0 \\ .3 & .3 & .2 & .05 & .05 & .1 & 0 & 0 \\ .15 & .15 & .15 & .15 & .15 & .15 & .1 & 0 \\ .125 & .125 & .125 & .125 & .125 & .125 & .125 & .125 \\ .3 & .1 & .1 & .1 & .1 & .1 & .1 & .1 \end{bmatrix}$$

$$P^3 = P^6 = \begin{bmatrix} .8 & .2 & 0 & 0 & 0 & 0 & 0 & 0 \\ .4 & .3 & .3 & 0 & 0 & 0 & 0 & 0 \\ .2 & .25 & .25 & .3 & 0 & 0 & 0 & 0 \\ .1 & .24 & .25 & .31 & .1 & 0 & 0 & 0 \\ .2 & .2 & .2 & .1 & .1 & .2 & 0 & 0 \\ .2 & .3 & .1 & .05 & .05 & .15 & .15 & 0 \\ .2 & .2 & .1 & .1 & .1 & .1 & .1 & .1 \\ .125 & .125 & .125 & .125 & .125 & .125 & .125 & .125 \end{bmatrix}$$

Second, the probability matrices that characterize the probability distribution $PD2$ are given below.

$$P^i = \begin{bmatrix} .5 & .5 & 0 & 0 & 0 & 0 & 0 & 0 \\ .4 & .3 & .3 & 0 & 0 & 0 & 0 & 0 \\ .25 & .25 & .25 & .25 & 0 & 0 & 0 & 0 \\ .2 & .2 & .2 & .2 & .2 & 0 & 0 & 0 \\ .1667 & .1667 & .1667 & .1667 & .1667 & .1665 & 0 & 0 \\ .1429 & .1429 & .1429 & .1429 & .1429 & .1429 & .1426 & 0 \\ .125 & .125 & .125 & .125 & .125 & .125 & .125 & .125 \\ .125 & .125 & .125 & .125 & .125 & .125 & .125 & .125 \end{bmatrix},$$

for $i = 1, \cdots, 6$.

Third, the probability matrices that characterize probability distribution $PD3$ are given below.

$$P^1 = P^4 = \begin{bmatrix} .2 & .8 & 0 & 0 & 0 & 0 & 0 & 0 \\ .35 & .1 & .55 & 0 & 0 & 0 & 0 & 0 \\ .25 & .25 & .25 & .25 & 0 & 0 & 0 & 0 \\ .03 & .33 & .21 & .06 & .37 & 0 & 0 & 0 \\ .1 & .15 & .05 & .25 & .2 & .25 & 0 & 0 \\ .1 & .15 & .15 & .1 & .1 & .2 & .2 & 0 \\ .125 & .025 & .025 & .025 & .025 & .025 & .125 & .625 \\ .2 & 0 & 0 & .1 & .1 & .1 & .1 & .4 \end{bmatrix}$$

$$P^2 = P^5 = \begin{bmatrix} .3 & .7 & 0 & 0 & 0 & 0 & 0 & 0 \\ .25 & .1 & .65 & 0 & 0 & 0 & 0 & 0 \\ .15 & .05 & .35 & .45 & 0 & 0 & 0 & 0 \\ .225 & .125 & .125 & .525 & 0 & 0 & 0 & 0 \\ .1 & 0. & .2 & .05 & .05 & .6 & 0 & 0 \\ .1 & .1 & 0 & 0 & .15 & .15 & .5 & 0 \\ .125 & .025 & .025 & .025 & .025 & .025 & .125 & .625 \\ .05 & .05 & .05 & .05 & .15 & .05 & .05 & .55 \end{bmatrix}$$

$$P^3 = P^6 = \begin{bmatrix} .4 & .6 & 0 & 0 & 0 & 0 & 0 & 0 \\ .2 & .1 & .7 & 0 & 0 & 0 & 0 & 0 \\ .2 & .25 & .1 & .45 & 0 & 0 & 0 & 0 \\ .05 & .24 & 0 & .31 & .4 & 0 & 0 & 0 \\ .2 & .2 & .2 & .1 & .1 & .2 & 0 & 0 \\ .2 & .15 & .1 & .05 & .05 & .15 & .3 & 0 \\ .05 & .2 & .1 & .1 & .1 & .1 & .1 & .25 \\ .05 & .05 & .05 & .05 & .05 & .05 & .075 & .625 \end{bmatrix}.$$

References

[1] G. M. Baudet, "Asynchronous Iterative Methods for Multiprocessors," *J. ACM*, vol. 25, pp. 226-244, 1978.

[2] B. F. Beidas, Distributed Asynchronous Implementation with Stochastic Delays of Iterative Algorithms, Ph.D. Dissertation, University of Southern Calironia, Los Angeles, CA, 1992.

[3] B. F. Beidas and G. P. Papavassilopoulos, "Convergence Analysis of Linear Asynchronous Iterations with Stochastic Delays," to appear in *Parallel Computing*.

[4] D. P. Bertsekas, "Distributed Dynamic Programming," *IEEE Trans. Auto. Cont.*, vol. AC-27, pp. 610-616, 1982.

[5] D. P. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, New Jersey, 1987.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation, Numerical Methods*. Prentice Hall, 1989.

[7] D. Chazan and W. Miranaker, "Chaotic Relaxation," *Linear Algebra Appl.*, vol. 2, pp. 199-222, 1969.

[8] H. T. Kung, "Synchronized and Asynchronous Parallel Algorithms for Multiprocessors," in J. F. Traub, ed., *Algorithms and Complexity: New Directions and Recent Results*. Academic Press, pp. 153-200, 1976.

[9] H. J. Kushner and G. Yin, "Asymptotic Properties of Distributed and Communicating Stochastic Approximation Algorithms," *SIAM J. Cont. and Optimiz.*, vol. 25, pp. 1266-1290, 1987.

[10] E. Kaszkurewicz, A. Bhaya and D. D. Šiljak, "On the Convergence of Parallel Aynchronous Block-Iterative Computations," *Linear Algebra Appl.*, vol. 131, pp. 139-160, 1990.

[11] J. M. McQuillan, I. Richer and E. C. Rosen, "The New Routing Algorithm for the ARPANET," *IEEE Trans. Commun.*, vol. COM-28, pp. 711-719, 1980.

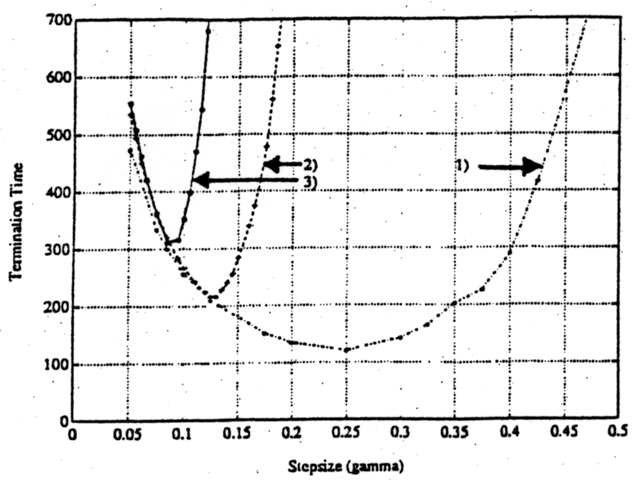
[12] D. D. Šiljak, *Large-Scale Dynamic systems*. North-Holland, New York, 1978.

[13] P. Tseng, D. P. Bertsekas and J. N. Tsitsiklis, "Partially Asynchronous, Parallel Algorithms for Network Flow and Other Problems," *SIAM J. Cont. and Optimiz.*, vol. 28, pp. 678-710, 1990.

[14] J. N. Tsitsikils and D. P. Bertsekas, "Distributed Asynchronous Optimal Routing in Data Networks," *IEEE Trans. Auto. Cont.*, vol. AC-31, pp. 325-332, 1986.

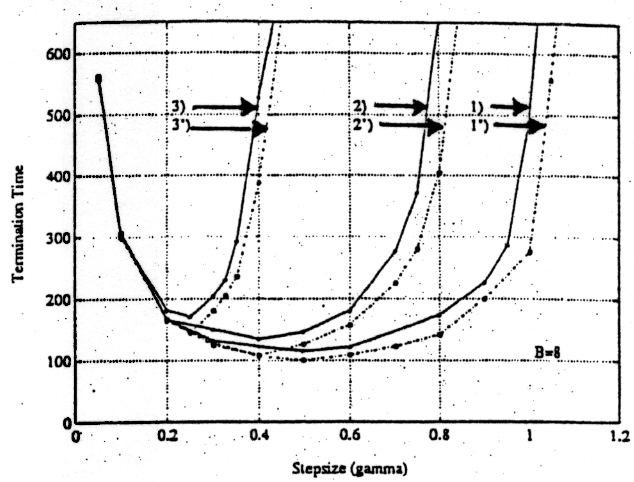
[15] J. N. Tsitsiklis, D. M. Bertsekas and M. Athans, "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," *IEEE Trans. Auto. Cont.*, vol. AC-31, pp. 803-812, 1986.

[16] A. Üresin, "Asynchronous Iterative Algorithms for Problems with Discrete Data," Ph.D. Thesis, University of Southern California, Los Angeles, CA, 1990.



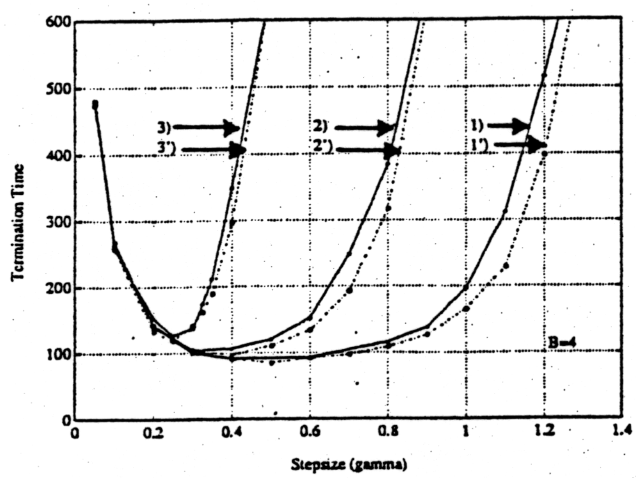
1) Distributed asynchronous algorithms with $B = 4$
2) Distributed asynchronous algorithms with $B = 8$
3) Distributed asynchronous algorithms with $B = 12$

Figure 2: Performance curves for distributed asynchronous algorithms with bounded delays for different communication delay bounds. 100 runs were considered.



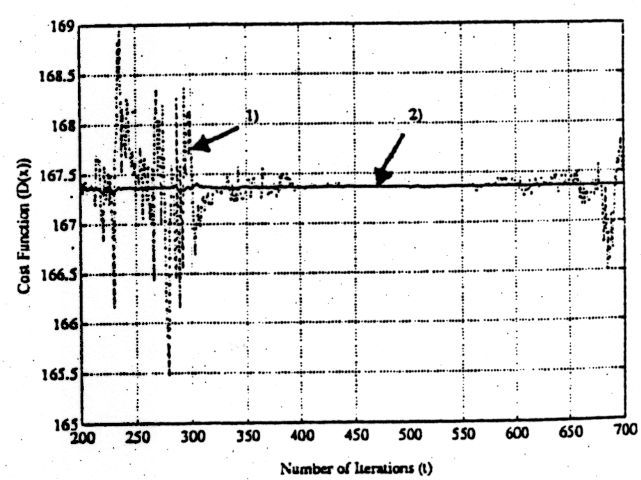
1) Algorithms with Markov delays for probability distribution $PD1$
1') Algorithms with independent delays whose probabilities equal the limiting behavior of $PD1$
2) Algorithms with Markov delays for probability distribution $PD2$
2') Algorithms with independent delays whose probabilities equal the limiting behavior of $PD2$
3) Algorithms with Markov delays for probability distribution $PD3$
3') Algorithms with independent delays whose probabilities equal the limiting behavior of $PD3$

Figure 4: Performance curves for distributed asynchronous algorithms with stochastic delays under different probability distributions. $B = 8$ and 1C0 runs were considered.



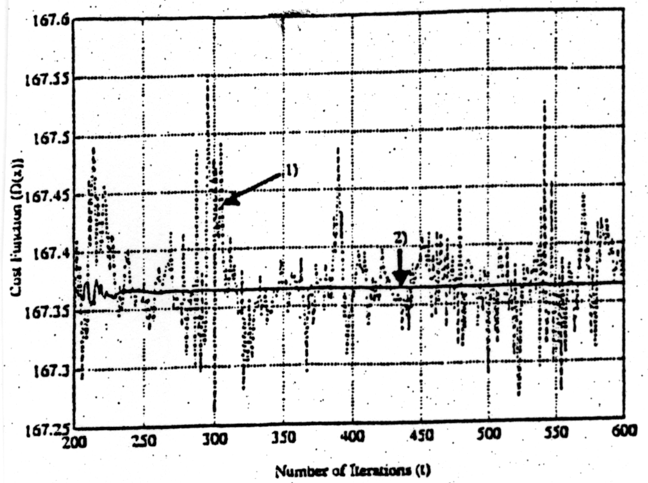
1) Algorithms with Markov delays for probability distribution $PD1$
1') Algorithms with independent delays whose probabilities equal the limiting behavior of $PD1$
2) Algorithms with Markov delays for probability distribution $PD2$
2') Algorithms with independent delays whose probabilities equal the limiting behavior of $PD2$
3) Algorithms with Markov delays for probability distribution $PD3$
3') Algorithms with independent delays whose probabilities equal the limiting behavior of $PD3$

Figure 3: Performance curves for distributed asynchronous algorithms with stochastic delays under different probability distributions. $B = 4$ and 100 runs were considered.



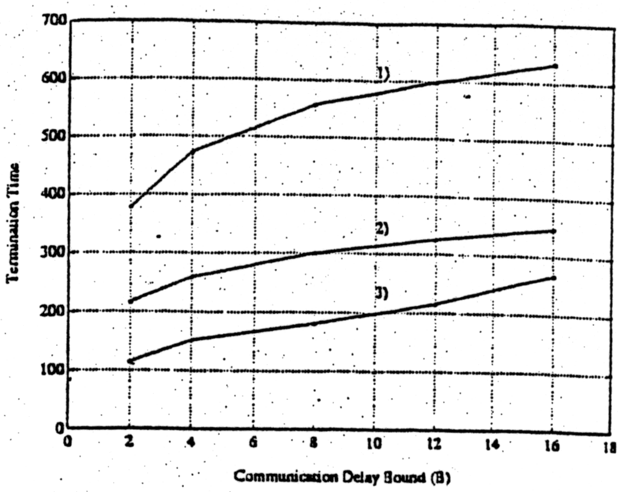
1) Algorithms with Markov delays
2) Algorithms with independent delays

Figure 5: Performance curves for distributed asynchronous algorithms with stochastic delays under $PD1$ for $\gamma = 1.095$. $B = 8$ and 1 run were considered.



1) Algorithms with Markov delays.
2) Algorithms with independent delays.

Figure 6: Performance curves for distributed asynchronous algorithms with stochastic delays under $PD2$ for $\gamma = 0.845$. $B = 8$ and 1 run were considered.



1) Algorithms with Markov delays for stepsize $\gamma = 0.05$
2) Algorithms with Markov delays for stepsize $\gamma = 0.1$
3) Algorithms with Markov delays for stepsize $\gamma = 0.2$

Figure 7: Effects of stepsize and delay bound on performance for distribut asynchronous algorithms with Markov delays. 100 runs were considered.