Implementation of Distributed Asynchronous Algorithms with Stochastic Delays for Solving Time Drifting Optimization Problems

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 minimizes a functional whose minimum drifts with time is discussed. The communication delays among the processors are assumed to be stochastic with Markovian character. We present conditions under which the mean square and almost sure convergence to the sought nonstationary solution are guaranteed.

1 Introduction

Solving large scale problems efficiently has engendered the rapid progress of parallel processing by which fast computation is achieved. Algorithms can be executed either synchronously or asynchronously in a multiprocessor system. In a synchronous parallel algorithm, the exchange of messages has to be synchronized at the completion of each update which may cause a significant performance degradation. Distributed asynchronous iterations, however, allow processors to operate while adhering to little coordination of computation and maintaining a low frequency of message propagation. The attractive prospects gained from the distributed asynchronous implementation in parallel processors are the reduction of processor idle periods, enhanced efficiency, resource sharing and load balancing [4, 9].

Distributed asynchronous implementation of iterative algorithms has recently been an active area of research [1, 4, 18]. The range of applications that these algorithms span is quite diverse. Symbolic computation, artificial intelligence [18] and computer networks [4, 15] have been influential in shaping the development of distributed asynchronous iterations. Our scope of interest is in the numerical computations related to the solution of general systems of equations and optimization problems. Chazan and Miranaker [5] introduced a distributed asynchronous iteration to solve linear systems of equations which were then called chaotic iterations. Baudet [1] obtained a convergence condition for nonlinear problems. Tsitsiklis, Bertsekas and Athans [16] presented the initial work of the current investigation in which they formulated and proved the convergence of distributed asynchronous iterative gradient algorithms to solve problems of minimizing a certain cost function. Kaszkurewicz, Bhaya and Šiljak [8] implemented distributed asynchronous iterations to solve a class of nonlinear problems and derived results that retained the quasi-dominance conditions previously studied by Šiljak [13] for the synchronous case. In Beidas and Papavassilopoulos [3], a general linear model of distributed asynchronous iterations was studied, the communication delays of which were Markovian. Under these assumptions, a number of sufficiency conditions for convergence were developed that do not require the usual nonexpansiveness conditions contained in [4, 14]. Almost all of the numerous works addressed to minimize cost functions (e.g., [10]–[12]) are confined to dealing with locating minima that do not change with time. In that respect, Polyak and Tsypkin [12] developed a unified approach to problems that rely on the pseudogradient notion. The first work that studied algorithms with drift, i.e., where the minimum changes with time was first introduced by Dupac [6], where a method of stochastic approximation was treated under a condition of drift that is close to linear. Tsypkin, Kaplinskiy and Larionov [17] developed a more general type of adaptive approach in training under nonstationary conditions. The work of [6, 17] deal with the synchronous computation models.

In this paper, we present an algorithm of distributed asynchronous iterations that minimizes a function when the sought minimum no longer remains stationary but is rather a vector that shifts in time. This may be attributed to the drift in the characteristics of systems which renders the optimality condition time variant. Cases of interest are pattern recognition problems where the probabilistic characteristics evolve with time [7]. Another assumption used in devising the algorithm in this paper is that the communication delays among cooperative processors are stochastic. The stochastic delays are pertinent when the intercommunication network that the processors use behaves unpredictably due to, for instance, varying load conditions, temporary failure of some of its links or overloading a certain number of its nodes. Other cases that incorporate the Markovian stochastic delay assumption are achieved by imposing a reasonable restriction on our models where we assume that the order of the received information is preserved according to the manner it was generated. This will enable the number of transmitted messages of the models with stochastic delays to remain under control.

The structure of the paper is as follows. In Section 2 we provide a description of distributed asynchronous algorithm and using an example give a comparison against its synchronous counterpart. In Section 3 we introduce distributed asynchronous algorithms that treat nonstationary optimization problems, i.e., problems where the sought solution is a function of time. In Section 4 we discuss their convergence analysis. A comparison between our results and the ones contained in [17] for the synchronous case is given in Section 4.1. Finally, conclusions and possible extensions are listed in Section 5. The proofs of our result can be found in [2].

2 Description of Distributed Asynchronous Iterations

Consider a system of n processors that is utilized to solve a fixed point problem. Suppose that each processor i updates the variable x_i, where $x_i \in \mathcal{R}^{n_i}$ according to

$$x_i = f_i(x_1, x_2, \ldots, x_n), \text{ for } i = 1, 2, \ldots, n, \tag{1}$$

where the initial values for this problem are provided.

Implementing a synchronous algorithm of equation (1) requires that processor i can not proceed in its computation at time t unless the arrival of the updates from all other processors at time $t-1$ is attained. Figure 1 depicts the behavior of system of n processors where processor i upon updating x_i must remain idle while waiting to receive the messages from the others. In particular, the sluggish communication would impede the progress of the entire computation. It can also be shown that at synchronization points, processors have to wait for the slowest processor.

To overcome this difficulty, we resort to distributed asynchronous iterations where the close attention that should be paid to receiving the most recent information is downplayed. Hence, in this case after a processor completes its computation it broadcasts its information and starts on the next update with any available data. Figure 2 depicts the behavior of the distributed asynchronous algorithm that a system of n processors carries out. In essence, the idle periods encountered in the synchronous case are dispensed with at the possible expense of reducing the quality of computation. Here, we allow the communication delays to be stochastic in order to gain those merits discussed in the previous section and our intention next is to evaluate conditions under which the convergence of these distributed asynchronous algorithms is maintained.

3 System Model

We employ a model of n processors working asynchronously on minimizing a function, the minimum of which exists under conditions of time drift. Let $F(t,x)$ be a continuously differentiable cost function to be minimized. Notice that the optimum of $F(t,x)$ with respect to x changes with time and is written as $x^*(t)$. Assume that the trajectory of the minimum is described by the difference equation

$$x^*(t+1) = R(t, x^*(t)), \tag{2}$$

where $R(t,.)$ is a known nonlinear transformation and the initial value $x^*(0)$ is unknown. If the initial value $x^*(0)$ is known, then the minimization problem becomes superfluous as equation (2) can be explicitly evaluated. We, therefore, expect the arbitrariness of the initial value to stress the errors in tracking the minimum at the early stages of the algorithm. The assumption of parameterizing the drift was also made by Dupac [6] and Tsypkin, Kaplinskiy and Larionov [17].

We let $x_i(t) \in \mathcal{R}^{n_i}$, where $\sum_{i=1}^{n} n_i = N$ and allow each processor i to update $x_i(t)$. We denote $d_{ji}(t)$ as the delay incurred by transmitting a message from processor j to processor i at time t. We let the communication delays $\{d_{ji}(t)\}$, for all j and i, be stationary Markov chains with state space

$$S = \{1, 2, \ldots, B\},$$

where B is the maximum allowable communication delay for the transmitted messages. We let the probability transition matrix corresponding to $d_{ji}(t)$ be $P_{ji} = (p_{ji}(l,m))$, i.e.,

$$p_{ji}(l,m) = \Pr\{d_{ji}(t) = m \mid d_{ji}(t-1) = l\}$$

$$\text{for } l, m = 1, 2, \ldots, B, \quad (3)$$

where here and in the sequel $\Pr\{C\}$ denotes the probability of event C.

As a result of the communication delays we expect that the processors are unable to obtain the most recent information from the others. In particular, processor i has knowledge of the vector $y^i(t)$

$$y^i(t) = \begin{bmatrix} x_1(t+1-d_{1i}(t)) \\ \vdots \\ x_n(t+1-d_{ni}(t)) \end{bmatrix}. \quad (4)$$

We consider an algorithm in which the processors perform the updating according to the recursive scheme

$$x_i(t+1) = R_i(t, y^i(t)) - \gamma(t) s_i(t, R(t, y^i(t))). \quad (5)$$

Convergence is studied with the use of the Lyapunov function defined as the squared norm of the distance away from the desired minimum, i.e.,

$$V(t,x) = \frac{1}{2}\|x - x^*(t)\|^2, \quad (6)$$

where here and in the sequel $\|\cdot\|$ is the Euclidean norm.

Let $\mathcal{I}_t$ define the previous information of the algorithm until time t such that

$$\mathcal{I}_t = \{d_{ji}(\tau), s_i(\tau, R(\tau, y^i(\tau))) \\ , \text{for } \tau < t \text{ and } j, i = 1, \ldots, n\}. \quad (7)$$

$\mathcal{I}_0$ includes the initial condition information. We note that $x_i(t)$ is uniquely determined by the random variables defined by $\mathcal{I}_t$.

The basic assumptions are introduced, the form of which is expressed in terms of $y^i(t)$ which is the information available to each processor. This permits the individual verification of the basic assumptions by the various processors. It is important to note that the ability of such verification is an intrinsic property of distributed asynchronous iterations.

Basic Assumptions:

1. There exists deterministic positive $K_1(t)$ such that for all i, we have

$$E\left[(y_i^i(t) - x_i^*(t))' s_i(t, y^i(t)) \mid \mathcal{I}_t\right] \\ \geq K_1(t) E\left[\|y^i(t) - x^*(t)\|^2 \mid \mathcal{I}_t\right]. \quad (8)$$

2. There exist deterministic nonnegative $K_2(t)$ and $K_3(t)$ such that for all i, we have

$$E\left[\|s_i(t, y^i(t))\|^2 \mid \mathcal{I}_t\right] \\ \leq K_2(t) + K_3(t) E\left[\|y^i(t) - x^*(t+1)\|^2 \mid \mathcal{I}_t\right]. \quad (9)$$

3. There exist nonnegative $\alpha(t)$ and $\beta(t)$ such that

$$(1+\alpha(t))\|x(t) - x^*(t)\| \\ \leq \|R(t, x(t)) - R(t, x^*(t))\| \leq \\ (1+\beta(t))\|x(t) - x^*(t)\|. \quad (10)$$

4. For the initial approximation, we have

$$E\|x(0) - x^*(0)\|^2 < \infty \text{ and } \|x^*(0)\| < \infty. \quad (11)$$

Given the previous history of the algorithm, inequality (8) requires that the expected direction of $-s(t, y^i(t))$ is one of decrease with respect to the Lyapunov function $V(t, y^i(t))$ while inequality (9) places growth conditions on the update $s(t, y^i(t))$, respectively. The inclusion of $K_2(t)$ in inequality (9) is indicative of the presence of additive noise with variance that is not necessarily finite.

A special case of the system defined in equation (5) is the gradient algorithm where the update consists of the distorted version of the gradient, i.e.,

$$s_i(t, y^i(t)) = \nabla_i F(t, y^i(t)) + \zeta_i(t). \quad (12)$$

Here, the measurement error $\zeta_i(t)$ is an independent identically distributed random vector. Other cases can involve a scaled form of the gradient algorithms where $D(t)$ is an invertible scaling matrix such that

$$s_i(t, y^i(t)) = (D(t))^{-1} \nabla_i F(t, y^i(t)) + \zeta_i(t). \quad (13)$$

4 Convergence Analysis

We formulate the main convergence results of process (5). Let us denote

$$q(t) = (1+\beta(t))^2 + \gamma^2(t) K_3(t)(1+\beta(t))^2 \\ - 2\gamma(t) K_1(t)(1+\alpha(t))^2. \quad (14)$$

We say that a sequence $\nu(t)$ of random variables converges to a random variable ν almost surely if

$$\Pr\left\{\lim_{t\to\infty} \nu(t) = \nu\right\} = 1. \quad (15)$$

The analysis is carried out in the sense of exploiting the proper Lyapunov function that is based on the reduction of the error defined in equation (6) and moulding this error equation to fit the form of an easily manageable vector inequality.

Theorem 1 *Consider the sequence $\{x_i(t)\}$ generated by equation (5). Suppose that the cost function $F(t,x)$ has a unique minimum at $x = x^*(t)$ for any t. Let the basic assumptions (1) - (4) be satisfied. In addition, assume that*

1. $\sum_{t=0}^{\infty} q(t) < \infty$, $q(t) \geq 0$,

2. $\sum_{t=0}^{\infty} \beta(t) < \infty$,

3. $\sum_{t=0}^{\infty} \|R(t,0)\|^2 < \infty$,

4. $\sum_{t=0}^{\infty} \gamma^2(t) K_2(t) < \infty$.

Then for every initial condition the sequence $\{x_i(t)\}$ *converges to* $x_i^*(t)$ *in the mean square and almost surely for all* i.

Proof: See [2].

It can be shown that the condition needed for almost sure convergence, i.e., $\sum_{t=0}^{\infty} \gamma^2(t) K_2(t) < \infty$ is more stringent than $\lim_{t \to \infty} \gamma^2(t) K_2(t) = 0$ which is required for convergence in the mean square.

4.1 Comparison with the Synchronous Case

A little reflection is needed to show the underlying relationship between our results and those obtained for the case of synchronous iterations. The specialization of our model represented by equation (5) to the synchronous case indicates that

$$\Pr\{d_{ji}(t) \neq 1\} = 0, \text{ for all } j,\ i,\ t \quad (16)$$

and, therefore,

$$y^i(t) = x(t). \quad (17)$$

Tsypkin, Kaplinskiy and Larionov [17] considered the synchronous version of the gradient algorithm that minimizes a cost function under nonstationary conditions where from equations (12) and (17), we obtain

$$x(t+1) = R(t, x(t)) - \gamma(t) \left(\nabla F(t, R(t, x(t))) + \zeta(t)\right). \quad (18)$$

The sufficient conditions for convergence in the mean square of the synchronous iterations represented by equation (18) are the following:

1. for $q(t)$ defined in equation (14) such that $0 \leq q(t) < 1$, it holds

$$\sum_{t=0}^{\infty} (1 - q(t)) = \infty, \quad (19)$$

2.

$$\lim_{t \to \infty} \frac{\gamma^2(t) K_2(t)}{1 - q(t)} = 0. \quad (20)$$

For our distributed asynchronous model as $q(t) < 1$, conditions (1) and (4) of Theorem 1 are stricter than and imply those given by equations (19) and (20). We, therefore, realize that resorting to the distributed asynchronous schemes may be accomplished at the expense of fulfilling a stronger criterion of convergence.

5 Conclusions

We have proposed and investigated the behavior of distributed asynchronous iterations with stochastic delays that solve optimization problems with nonstationary minima. The plausibility of the notion of stochastic delays stems from the fact that it models the case of an unpredictable communication network among the processors and therefore addresses various reliability aspects and probabilistic load descriptions of the communication network. The analysis that establishes the sufficiency conditions required to guarantee mean square and almost sure convergence is based upon utilizing a Lyapunov function that reduces the distance away from the minimum point defined in equation (6) and showing that the adverse effects inflicted by the communication delays are negligible.

This work was further extended [2] to implement our model of distributed asynchronous iterations with stochastic delays in solving constrained optimization under conditions of time drift and to invoke the gained machinery in different applications. Furthermore, with the aid of simulation studies we evaluated the performance of distributed asynchronous iterations with stochastic delays and obtain comparison results with their synchronous counterparts.

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 California, Los Angeles, CA, 1992.

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

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

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

[6] V. A. Dupac, "A Dynamic Stochastic Approximation Method," *Ann. Math. Statistics*, vol. 36, pp. 1695-1702, 1965.

[7] R. Figueiredo, "Convergence Algorithms for Pattern Recognition in Nonlinearly Evolving Nonstationary Environment," *Proc. IEEE 56*, pp. 188-189, 1968.

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

[9] 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.

[10] D. G. Luenberger, *Linear and Nonlinear Programming*. Addison-Wesley Publishing Co., 1984.

[11] B. T. Polyak, "Convergence and Convergence Rate of Iterative Algorithms, Part I: General Case," *Automation Remote Control*, vol. 37, pp. 83-94, 1976.

[12] B. T. Polyak and Ya. Z. Tsypkin, "Pseudogradient Adaptation and Training Algorithms," *Automation Remote Control*, vol. 34, pp. 45-68, 1973.

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

[14] 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.

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

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

[17] Ya. Z. Tsypkin, A. I. Kaplinskiy and K. A. Larionov, "Adaptation and Learning Algorithms under Nonstationary Conditions," *Engineering Cybernetics*, vol. 5, pp. 829-840, 1970.

[18] A. Üresin and M. Dubois, "Parallel Asynchronous Algorithms for Discrete Data," *J. ACM*, vol. 37, pp. 588-606, 1990.

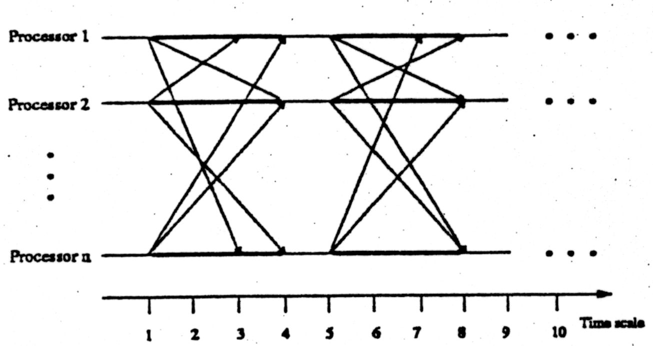

Figure 1: Depiction of typical behavior of the synchronous implementation of a system of n processors where the computation time is one unit. Note that the heavily darken lines indicate the idle periods that the processors encounter in the synchronous implementation.

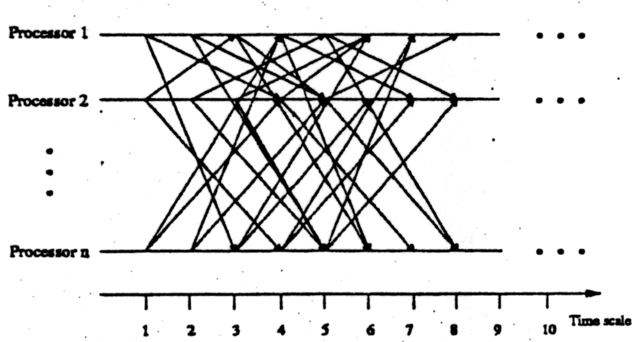

Figure 2: Depiction of typical behavior of the distributed asynchronous implementation of a system of n processors where the computation time is one unit. Note that the idle periods are dispensed with at the possible expense of reducing the quality of computation.