November, 1983 LIDS-P-1344

DISTRIBUTED OPTIMIZATION ALGORITHMS WITH COMMUNICATIONS*

by

John Tsitsiklis and Michael Athans
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

We discuss the convergence properties of asynchronous distributed iterative optimization algorithms, tolerating communication delays. We focus on a gradient-type algorithm for minimizing an additive cost function and present sufficient conditions for convergence. We view such an algorithm as a model of adjustment of the decisions of decision makers in an organization and we suggest that our results can be interpreted as guidelines for designing the information flows in an organization.

1. Introduction

This paper concerns the convergence properties and communication requirements of asynchronous distributed optimization algorithms, tolerating communication delays. The results being presented may be interpreted as pertaining to the performance of potential parallel computing machines. Alternatively, an approach which we pursue in this paper, our results may be viewed as a description of the adjustment process in a distributed organization, possibly involving human decision makers. Moreover, it could be maintained that the mathematical models discussed here, capture some aspects of the ever-present "bounded rationality" of human decision makers [Simon, 1980].

Our motivation is the following: A *boundedly rational* decision maker solving an optimization problem (minimize J(x)), may be viewed as an iterative optimization algorithm, whereby a tentative decision x(n) is made at time n, and then the decision is updated, in a direction of improvement. For example, we may have

$$x(n+1) = x(n) - \partial \frac{\partial J}{\partial x} (x(n)), \qquad (1.1)$$

which corresponds to the well-known gradient algorithm. By extending the above analogy to more complex settings, an *organization* (or, at least, some aspects of it) consisting of *cooperative*, boundedly rational decision makers may be viewed as a distributed algorithm. For example, suppose that x is a decision vector and that the i-th decision maker is in charge of the i-th component x_i of x, which he updates according to

$$x_i(n+1) = x_i(n) - \partial_i \frac{\partial J}{\partial x_i} (x(n)) \; . \qquad (1.2)$$

If each decision maker was to update his part of the decision (his own component), at each instance of time according to (1.2), we would effectively a *synchronous* distributed implementation of the centralized gradient algorithm. Synchronous algorithms have been studied in a variety of contexts [Arrow and Hurwicz, 1960; Gallager, 1977] but they also have certain *drawbacks*:
(1) Decision maker i, in order to update $x_i(n)$ according to (1.2), he needs to know x(n), at time n. This requires that each decision maker informs all others, at each time instance, on the adjustments of his decisions. So, in some sense, the synchronous model requires "a lot of communications."
(2) A second drawback of synchronous algorithms is that communication delays can introduce bottlenecks and slow down the algorithm. In particular, the time between two consecutive updates has to be at least as large as the *maximum* communication delay between any pair of decision makers.
(3) Finally, complete synchronization is certainly an unrealistic model of human organizations.

For the above reasons, we choose to study asynchronous distributed versions of iterative optimization algorithms, in which decision makers do not need to communicate to every other decision maker at each time instance. Such algorithms avoid communication overloads, they are not excessively slowed down by communication delays and there is not even a requirement that each decision maker updates his decision at each time instance, which makes them even more realistic.

2. General Properties and Convergence Conditions of Asynchronous Distributed Algorithms

We now discuss the main principle underlying the class of asynchronous algorithms which we consider: as we mentioned, in Section 1, for a synchronous algorithm, each decision maker needs to be informed of the most recent value of the decisions of all other decision makers. Suppose now that decision maker i, at time n, needs for his computations the current value $x_j(n)$ of the j-th component of x, but he does not know this value. We then postulate that decision maker i will carry out his computations as in the synchronous algorithm, except that (not knowing $x_j(n)$) he will use the value of x_j in the most recent message he has received from decision maker j. Due to asynchronism and communication delays, decision maker i will, in general, use out-dated values of x_j to update his own decisions. However, updates based on out-dated information may be substantially better than not updating at all. The crucial questions which arise are: How much out-dated information may be tolerated? How frequent should communications be, so that the distributed algorithm operates in a desirable manner?

Questions of this nature have been addressed by Bertsekas [1982,1983] for the distributed version of the successive approximations algorithm for dynamic programming and the distributed computation of fixed points. We have obtained general convergence results of a related nature for the asynchronous distributed versions of deterministic and stochastic iterative pseudo-gradient [Poljak and Tsypkin, 1973] (or "descent-type") algorithms. Some representative algorithms covered by our general results are deterministic

* Research supported by ONR under Contract ONR/N00014-77-C-0532 (NR-041-519).

gradient-type algorithms, as well as stochastic approximations algorithms. Due to space considerations, we only discuss here the nature of the results. Exact statements and the proofs may be found in [Tsitsiklis, 1983] and in forthcoming publications. Preliminary versions of these results appear in [Tsitsiklis, Bertsekas and Athans, 1983].

To discuss the nature of the convergence conditions, we distinguish two cases:

A. Constant Step-Size Algorithms (e.g. gradient algorithm)

For such algorithms it has been shown that convergence to the centralized optimum is obtained, provided that the time between consecutive communications between pairs of decision makers, plus the communication delay, is bounded by an appropriate constant. Moreover, the larger the step-size (i.e. the constant ∂ in equation (1.2)), the smaller the above mentioned constant. The latter statement admits the appealing interpretation that the larger the updates by each decision maker, the more frequent communications are required.

B. Decreasing Step-Size Algorithms (e.g. stochastic approximation algorithms)

In this case, the algorithm becomes slower and slower as the time index increases. This allows the process of communications to become progressively slower, as well. In particular, it has been shown that convergence to the centralized optimum is obtained even if the time between consecutive communications between pairs of decision makers, plus communication delays, increase without bound, as the algorithm proceeds, provided that the rate of increase is not too fast.

3. A Distributed Gradient Algorithm

In this section we consider a rather simple distributed algorithm for minimizing an additive cost function. Due to the simplicity of the algorithm, we are able to derive convergence conditions which are generally tighter than the general conditions discussed in the previous sections. It will be seen shortly, that these conditions admit appealing organizational interpretations.

The conceptual motivation behind our approach is based on the following statement:

> If an optimization problem consists of subproblems, each subproblem being assigned to a different decision maker, then the frequency of communications between a pair of decision makers should reflect the degree by which their subproblems are coupled.

The above statement is fairly hard to capture mathematically. This is accomplished, however, to some extent, by the model and the results of this section.

Let $J: R^M \to R$ be a cost function to be minimizedm with a special structure:

$$J(x) = J(x_1,\dots,x_M) = \sum_{i=1}^{M} J^i(x_1,\dots,x_M) \qquad (3.1)$$

where $J^i: R^M \to R$. So far, equation (3.1) does not impose any restriction on J; we will be interested, however, in the case where, for each i, J^i depends on x_i and only a few more components of x; consequently, the Hessian matrix of each J^i is sparse.

We view J^i as a cost directly faced by the i-th decision maker. This decision maker is free to fix or update the component x_i, but his cost also depends on a few interaction variables (other components of x) which are under the authority of other decision makers.

We may visualize the structure of the interactions by means of a directed graph $G=(V,E)$:

(i) The set V of nodes of G is $V=\{1,\dots,M\}$

(ii) The set of edges E of the graph is

$$E=\{(i,j): J^j \text{ depends on } x_i\} \qquad (3.2)$$

Since we are interested in the fine structure of the optimization problem, we quantify the interactions between subproblems by assuming that the following bounds are available:

$$\left|\frac{\partial^2 J}{\partial x_i \partial x_j}\right| \le K_{ij}, \quad \left|\frac{\partial^2 JK}{\partial x_i \partial x_j}\right| \le K^k_{ij}, \quad \forall x \in R^M, \qquad (3.3)$$

where (without loss of generality)

$$K_{ij} \le \sum_{k=1}^{M} K^k_{ij}\,. \qquad (3.4)$$

A synchronous distributed gradient-type algorithm for this problem could be:

1. For each $(i,j) \in E$, decision maker j evaluates

$$\lambda^j_i(n) = \frac{\partial J^j}{\partial x_i}(x(n)) \qquad (3.5)$$

2. For each $(i,j) \in E$, decision maker j transmits $\lambda^j_i(n)$ to decision maker i.
3. Each decision maker i updates x_i according to

$$x_i(n+1) = x_i(n) - \partial_i \sum_{j=1}^{M} \lambda^j_i(n) \qquad (3.6)$$

4. For each $(i,j) \in E$, decision maker i transmits $x_i(n+1)$ to decision maker j.

We now consider the asynchronous version of the above algorithm. Let $x^i(n)=(x^i_1(n),\dots,x^i_M(n))$ denote a decision vector (element of R^M) stored in the memory of decision maker i at time n. We also assume that each decision maker i stores in his memory another vector $(\lambda^1_i(n),\dots,\lambda^M_i(n))$ with his estimates of $\frac{\partial J^1}{\partial x_i},\dots,\frac{\partial J^M}{\partial x_i}$. Unlike the synchronous algorithm, we do not require that a message be transmitted at each time stage and we allow communication delays. So let:

$p^{ki}(n)$ = the time that a message with a value of x_k was sent from processor k to processor i, and this was the last such message received no later than time n.

$q^{ki}(n)$ = the time that a message with a value of $\frac{\partial J^k}{\partial x_i}$ was sent from processor k to processor i, and this was the last such message received no later than time n.

For consistency of notation we let

$$p^{ii}(n) = q^{ii}(n)=n, \quad \forall_i, \forall_n\,. \qquad (3.7)$$

With the above definitions, we have:

$$x^i_k(n) = x^k_k(p^{ki}(n)), \quad \forall_n, \forall (k,i) \in E, \qquad (3.8)$$

$$\lambda^k_i(n) = \frac{\partial J^k}{\partial x_i}(x^k(q^{ki}(n))), \quad \forall_n, \forall (i,k) \in E. \qquad (3.9)$$

Equations (3.8), (3.9) together with

$$x_i^i(n+1) = x_i^i(n) - \partial_i \sum_{j=1}^{M} \lambda_i^j(n) \qquad (3.10)$$

specify completely the asynchronous distributed algorithm of interest.

Let us now assume that the time between consecutive communications and the communication delays are bounded. We allow, however, these bounds to be different for each pair of processors and each type of message:

Assumption: For some constants P^{ik}, Q^{ik},

$$n-P^{ik} \leq p^{ik}(n) \leq n, \qquad \forall (i,k) \in E, \ \forall_n, \qquad (3.11)$$

$$n-Q^{ik} \leq q^{ik}(n) \leq n, \qquad \forall (k,i) \in E, \ \forall_n, \qquad (3.12)$$

Note that we may let $P^{ii} = Q^{ii} = 0$.

The following result states that the algorithm converges if P^{ik} and Q^{ik} are not too large compared to the degree of coupling between different subproblems. [Tsitsiklis, 1983].

Theorem 3.1: Suppose that for each i

$$\frac{2}{J_i} > \sum_{j=1}^{M} K_{ij} + \sum_{k=1}^{M} \sum_{j=1}^{M} K_{ij}^k (P^{ik}+Q^{kj}+P^{jk}+Q^{ki}) \ . \qquad (3.13)$$

Let $z(n)=(x_1^1(n), x_2^2(n), \ldots, x_M^M(n))$. Then,

$$\lim_{n \to \infty} \frac{\partial J}{\partial x_i}(z(n))=0, \qquad \forall_{i,j} \cdot \qquad (3.14)$$

We close this section with a few remarks:

1. The bounds provided by (3.13) are sufficient for convergence but not necessary. It is known [Bertsekas, 1982b] that a decentralized algorithm of a similar type may converge in certain special cases, even if the ∂_i's are held fixed, while the bounds P^{ik}, Q^{ik} are allowed to be arbitrarily large. So, the gap between the sufficient conditions (3.13) and the necessary conditions may be substantial. Further research should narrow this gap.

2. The convergence rate of the distributed algorithm should be expected to deteriorate as the bounds P^{ik}, Q^{ik} increase. A characterization of the convergence rate, however, seems to be a fairly hard problem.

4. Towards Organizational Design

Suppose that we have a divisionalized organization and that the objective of the organization is to minimize a cost J which is the sum of the costs J^i faced by each division. To each division, these corresponds a decision maker which is knowledgeable enough about the structure of the problem he is facing, to the extent that given a tentative decision he is able to change his decision in a direction of improvement. Moreover, suppose that the divisions are interacting in some way; that is, the decision of one decision maker may affect the costs of another division. Suppose, finally, that decision makers regularly update their decisions taking into account the decisions of other decision makers and the effects of their own decisions on other divisions. Messages are being exchanged from time to time carrying the required information. Clearly, the mathematical model of Section 3 may be viewed as a model of the above situation.

A natural question raised by the above described situation concerns the design of the information flows within the organization, so as to guarantee smooth operation. But this is precisely the issue addressed by Theorem 3.1: the bounds K_{ij}^k may be thought as quantifying the degree of coupling between divisions; the bouns P^{ij}, Q^{ij} describe the frequency of communications and ∂_i represents the speed of adjustment. Theorem 3.1 links all these quantities together and provides some conditions for smooth operation, whereby communication rates are prescribed in terms of the degree of coupling.

We may conclude that the approach of Section 3 may form the basis of a procedure for designing an organizational structure, or -more precisely- the information flows within an organization. Of course, Theorem 3.1 does not exhaust the subject. In particular, Theorem 3.1 suggests a set of feasible organizational structures, with generally different convergence rates. There remains the problem of choosing a "best" such structure.

It is also conceivable that the structure of the underlying optimization problem slowly changes with time, and so do the bounds K_{ij}^k, but in a time scale slower than the time scale of the adjustment process. In such a case, the bounds P^{ij}, Q^{ij} should also change. This leads to a natural two-level organizational structure: At the lower level, we have a set of decision makers continuously adjusting their decisions and exchanging messages. At a higher level, we have a supervisor who monitors changes in K_{ij}^k and accordingly instructs the low-level decision makers to adjust their communication rates. Note that the supervisor does not need to know the details of the cost function; he only needs to know the degree of coupling between divisions. This seems to reflect the actual structure of existing organizations. Low level decision makers are "experts" on the problems facing them, while higher level decision makers only know certain structural properties of the overall problem and make certain global decision, e.g. setting the communication rates.

Event-Driven Communications

We now discuss a slightly different "mode of operation" for the asynchronous algorithm, which has also clear organizational implications. It should be clear that communications are required by the distributed algorithm so that decision makers are informed of changes occuring elsewhere in the system. Moreover, the bounds P^{ik}, Q^{ik} of Section 3 effectively guarantee that a message is being sent whenever a substantial change occurs. The same effect, however, could be accomplished without imposing bounds on the time between consecutive message transmissions: each decision maker could just monitor his decisions and inform the others whenever a substantial change occurs. It seems that the latter approach could result to significant savings in the number of messages being exchanged, but further research is needed on this topic.

5. Conclusions

A large class of deterministic and stochastic iterative optimization algorithms admit natural distributed asynchronous implementations. Such implementations (when compared to their synchronous counterparts) may retain the desired convergence properties, while reducing communication requirements and removing bottlenecks caused by communication delays.

We have focused on a deterministic gradient-type algorithm for an additive cost function and we have shown that the communication requirements depend in a natural way on the degree of coupling between different components of the cost function. This approach addresses the basic problem of designing the information flows in a distributed organization and may form the basis for a systematic approach to organizational design.

References

Arrow, K.J., L. Hurwicz, (1960), "Decentralized and Computation in Resource Allocation," in Essays in Economics and Econometrics, R.W. Pfouts, Ed., Univ. of North Carolina Press, Chapel Hill, N.C. pp. 34-104.

Bertsekas, D.P., (1982a), "Distributed Dynamic Programming," IEEE Transactions on Automatic Control, Vol. AC-27, No. 3, pp. 610-616.

Bertsekas, D.P., (1982b), "Distributed Computation of Fixed Points," Technical Report LIDS-P-1135, Laboratory for Information and Decision Systems, MIT; to appear in Mathematical Programming.

Gallager, R.G., (1977), "A Minimum Delay Routing Algorithm Using Distributed Computation," IEEE Transactions on Communications, Vol. COM-25, No. 1, pp. 73-85.

Poljak, B.T., Y.Z. Tsypkin, (1973), "Pseudogradient Adaptation and Training Algorithms," Automatika i Telemekhanika, No. 3, pp. 45-68.

Simon, H.A., (1980), "The Sciences of the Artificial, MIT Press, Cambridge, MA.

Tsitsiklis, J.N., (1983), "Problems in Distributed Decision Making," Technical Report, Laboratory for Information and Decision Systems, MIT, in preparation.

Tsitsiklis, J.N., Bertsekas, D.P., Athans, M., (1983), "Distributed Deterministic and Stochastic Optimization Algorithms with Applications in System Identification," Proceedings of the 1983 Melecon Conference, Athens, Greece.