Authors study asynchronous distributed iterative optimization algorithms.

Several processors perform computations and exchange messages with the end goal of minimizing a certain cost function.

Under certain assumptions, asynchronous distributed algorithms have similar convergence properties as their centralized counterparts.

Constant step-size algorithms: Time between consecutive communications has to be bounded for convergence to be guaranteed.

Decreasing step-size algorithms: Convergence is guaranteed even if the time between consecutive communications increases without bound as algorithm proceeds.
Let \( \{1, 2, \ldots, M\} \) be the set of processors that participate in the distributed computation.

Let \( H_1, H_2, \ldots, H_L \) are finite dimensional real vector spaces.

Let \( H = H_1 \times H_2 \times \ldots H_L \) be endowed with the Euclidean norm.

Each processor stores an element of \( H \) in its memory.

Let \( x^i(n) \in H \) be the value stored by the \( i^{\text{th}} \) processor at time \( n \).

Let \( s^i(n) \in H \) be the “step” computed by processor \( i \) at time \( n \) to be used in evaluating \( x^i(n + 1) \).
Processor $i$ may transmit some or all of the components of $\chi^i(n)$ to some (or all or none) of the other processors.

If a message is received by processor $i$ from processor $j$ at time $n$ containing an element of $H_l$, let $t_{ij}^l(n)$ denote the time that this message was sent.

Let $T_{ij}^l$ be the set of all times that processor $i$ receives a message from $j$, containing an element of $H_l$.

Assume that $T_{ij}^l$ is either infinite or empty.

Objective: To minimize a nonnegative cost function $J : H \to [0, \infty)$.

Let $\gamma^i(n)$ be non negative scalar step sizes.
Componentwise update to obtain $x^i(n + 1) \in \mathbb{H}$ is given by:

$$x^i(n + 1) = \sum_{j=1}^{M} a_{ij}^l(n) x^i_l(j^l(n)) + \gamma^i(n) s_i^i(n), \quad n \geq 1 \quad (2.1)$$

where $s_i^i(n)$ is the $l$th component of $s^i(n)$ and the coefficients $a_{ij}^l(n)$ are scalars satisfying

i) $a_{ij}^l(n) \geq 0 \quad \forall \ i, j, l, n,$ \hspace{1cm} (2.2)

ii) $\sum_{j=1}^{M} a_{ij}^l(n) = 1, \quad \forall \ i, l, n,$ \hspace{1cm} (2.3)

iii) $a_{ij}^l(n) = 0, \quad \forall \ n \notin T_{ij}, i \neq j.$ \hspace{1cm} (2.4)

$a_{ij}^l(n)$ are called combining coefficients.

Evaluation and addition of $\gamma^i(n)s_i^i(n)$ in equation (2.1) is referred to as performing a "computation".
Let $T^i_l$ denote the set of all times that processor $i$ performs a computation involving $l^{th}$ component.

Whenever $n \notin T^i_l$, $s^i_l(n) = 0$.

$T^i_l$ is either infinite or empty.

Accordingly the processor $i$ for which $s^i_l(n)$ is non-zero is called a computing processor.

Let $G_l = (V, E_l)$ be a directed graph for each component $l \in \{1, \ldots, L\}$ with nodes $V = \{1, \ldots, M\}$ corresponding to set of processors.

An edge $(j, i)$ belongs to $E_l$ iff $T^{ij}_l$ is infinite.
Assumptions on the Delay and the Combining Coefficients

Assumption 2.1 (Strong Connectivity)

For each component $l \in \{1, \ldots, L\}$,

1. There is at least one computing processor for component $l$.
2. There is a directed path in $G_l$, for every computing processor to every other processor.
3. There is some $\alpha > 0$ such that:
   - If processor $i$ receives a message from processor $j$ at time $n$, then $a_{ij}^l(n) \geq \alpha$.
   - For every computing processor $i$, $a_{ii}^l \geq \alpha$, $\forall n$.
   - If processor $i$ is non-computing and has in degree (in $G_l$) large than or equal to 2, then $a_{ii}^l(n) \geq \alpha$. 
Assumptions on the Delay and the Combining Coefficients

**Assumption 2.2 (Strictly Bounded Intercommunication Intervals)**

The time between consecutive transmissions of component $\chi_i^j$ from processor $j$ to processor $i$ is bounded by some $B_1 \geq 0$ for all $(j, i) \in E_l$.

**Assumption 2.3 (Bounded Intercommunication Intervals)**

There are constants $B_1 > 0$, $\beta \geq 1$, such that, for any $(j, i) \in E_l$, and for any $n$, at least one message $\chi_i^j$ is sent from processor $j$ to processor $i$ during the time interval $[B_1 n^\beta, B_1 (n + 1)^\beta]$. Moreover, the total number of messages sent and/or received during any such interval is bounded.

**Assumption 2.4 (Bounded Delays)**

Communication delays are bounded by some $B_0 \geq 0$, i.e., $\forall i, j, l$ and $n \in T_{ij}^l$, we have $n - t_{ij}^l(n) \leq B_0$. 
Consensus Example: \(^1\) (Unbounded Intercommunication Intervals)

- Consider a set of 3 agents that exchange messages to come to an agreement (consensus) upon a scalar \(x\).
- No cost function to minimize.
- Let \(\bar{x}(n) = (x^1(n), x^2(n), x^3(n))\) where \(x^i(n)\) is the value stored by processor \(i\) at time \(n\).
- Let \(\bar{x}(0) = (0, 0, 1)\).

**Sequence of Exchanges**

- Agent 3 communicates to agent 1; and agent 1 forms an average of its own value and the received value.
- This is repeated \(t_1\) times, where \(t_1\) is large enough so that \(x^1(t_1) \geq 1 - \epsilon_1\) for some positive constant \(\epsilon_1\).
- Thus \(\bar{x}(t_1) \approx (1, 0, 1)\).

\(^1\)Exercise 3.1 in p.317 of *Parallel and Distributed Computation: Numerical Methods*, D. P. Bertsekas and J. N. Tsitsiklis
Consensus Example : (Unbounded Intercommunication Intervals)

Sequence of Exchanges

- Agent 2 communicates to agent 3, \( t_2 \) times, where \( t_2 \) is large enough so that \( x^3(t_1 + t_2) \leq \epsilon_1 \).
- Thus \( \bar{x}(t_1 + t_2) \approx (1, 0, 0) \).
- These two processes are repeated infinitely by replacing \( \epsilon_1 \) by \( \epsilon_k \) during the \( k^{th} \) repetition and permuting the agents at each repetition.
- Clearly \( t_1 \) and \( t_2 \) corresponding to \( \epsilon_k \) for each \( k \) are not bounded in any interval.
- After \( k \) repetitions, \( \bar{x}(t) \) will be within \( 1 - \epsilon_1 - \ldots - \epsilon_k \) of a unit vector.
- So if we choose \( \epsilon_k \) so that \( \sum_{k=1}^{\infty} \epsilon_k < 1/2 \), then asymptotic consensus will not be obtained.
By linearity, there exist scalars $\Phi_{ij}^l(n|k)$ for $n \geq k$, such that

$$x_i^j(n) = \sum_{j=1}^{M} \Phi_{ij}^l(n|0)x_i^j(1) + \sum_{k=1}^{n-1} \sum_{j=1}^{M} \gamma^j(k)\Phi_{ij}^l(n|k)s_i^j(k).$$

(2.6)

$\Phi_{ij}^l(n|k) \geq 0$ are determined by the sequence of transmission and reception times and the combining coefficients.

Lemma 2.1

1. $\sum_{j=1}^{M} \Phi_{ij}^l(n|k) \leq 1$, $\forall i, l, n \geq k$. 
Lemma 2.1

2 Under assumptions 2.1 and 2.4 and either 2.2 or 2.3, 
\[ \lim_{n \to \infty} \Phi_{i j}^{l}(n | k) \] exists, for any \( i, j, k, l \). The limit is independent of \( i \) and denoted by \( \Phi_{j}^{l}(k) \). Moreover, there is a constant \( \eta > 0 \) such that, if \( j \) is a computing processor for component \( l \), then 
\[ \Phi_{j}^{l}(k) \geq \eta, \forall k. \]

The constant \( \eta \) depends only on other constants in the assumptions.

3 Under assumptions 2.1, 2.2 and 2.4, there exist \( d \in [0, 1) \), \( B \geq 0 \) such that 
\[
\max_{i,j} |\Phi_{i j}^{l}(n | k) - \Phi_{j}^{l}(k)| \leq Bd^{n-k}, \quad \forall \, l, \, n \geq k. \tag{2.10}
\]
Lemma 2.1

Under the assumptions 2.1, 2.3, and 2.4, there exist $d \in [0, 1)$, $\delta \in (0, 1]$, $B \geq 0$ such that

$$\max_{i,j} |\Phi^i_j(n|k) - \Phi^i_j(k)| \leq Bd^{n\delta - k\delta}, \quad \forall \ l, \ n \geq k. \ (2.11)$$

- Proving convergence of $\Phi(n|k)$ is equivalent to proving convergence of a sequence of products of stochastic matrices.
- If all processors cease updating (set $s^i(n) = 0$) from some time on, they will asymptotically converge to a common limit as seen from the following equation,

$$x^i_j(n) = \sum_{j=1}^{M} \Phi^i_j(n|0)x^i_j(1) + \sum_{k=1}^{n-1} \sum_{j=1}^{M} \gamma^i(k)\Phi^i_j(n|k)s^i_j(k).$$

(2.6)
More Notation

- Let \( \Phi^j(k) = (\Phi^j_1(k), \ldots, \Phi^j_L(k)) \in H \).
- Define \( y(n) \in H \) by
  \[
  y(n) = \sum_{j=1}^{M} \Phi^j(0)x^j(1) + \sum_{k=1}^{n-1} \sum_{j=1}^{M} \gamma^j(k)\Phi^j(k)s^j(k) \quad (2.13)
  \]
- Note that \( y(n) \) is generated recursively by
  \[
  y(n+1) = y(n) + \sum_{j=1}^{M} \gamma^j(n)\Phi^j(n)s^j(n). \quad (2.14)
  \]
- The processors would asymptotically agree at \( y(n) \) if they were to stop computing but keep communicating and combining at a time \( n \).
- Iteration (2.14) is approximately the same as the centralized descent algorithm.
The expected direction of update conditioned upon the past history of the algorithm is a descent direction with respect to the cost function to be minimized.

Authors present convergence results for natural distributed asynchronous versions of pseudogradient algorithms.

**Initialization**: Let \( \{x^1(1), \ldots, x^M(1)\} \) be random with finite mean and variance.

**Update and step-size**: Let \( s^i(n) \) be random while \( \gamma^i(n) \) is deterministic.

**Combining coefficients**: \( a^{ij}_l(n) \) and sequence of transmission and reception times are deterministic.

**All random variables are defined on a probability space** \( (\Omega, \mathcal{F}, P) \).
Let \( \{F_n\} \) be an increasing sequence of \( \sigma \)-fields contained in \( F \) and describing the history of the algorithm up to time \( n \).

**Objective of the Algorithm:** To minimize a nonnegative cost function \( J : H \rightarrow [0, \infty) \).

**Assumption 3.1**

\( J \) is continuously differentiable and its derivative satisfies the Lipschitz condition

\[
\| \nabla J(x) - \nabla J(x') \| \leq K \| x - x' \|, \quad \forall \ x, \ x' \in H \tag{3.1}
\]
Assumption 3.2

The updates $s_i^i(n)$ of each processor satisfy

$$E \left[ \frac{\partial J}{\partial x_l} (x^i(n)) s_i^i(n) | F_n \right] \leq 0, \quad \text{a.s., } \forall i, l, n.$$  \hspace{1cm} (3.2)

This assumption states that each component of the updates is in a descent direction, when conditioned on the past history.

Assumption 3.3

For some $K_0 \geq 0$ and for all $i, l, n$,

$$E[\|s_i^i(n)\|^2] \leq -K_0 E \left[ \frac{\partial J}{\partial x_l} (x^i(n)) s_i^i(n) \right]$$

That is, the variance of updates goes to zero as the gradient of the cost function goes to zero.
Theorem 3.1

Let the assumptions 2.1, 2.2, 2.4, 3.1, 3.2 and 3.3 hold. Suppose also that \( \gamma^i(n) \geq 0 \) and that \( \sup_{i,n} \gamma^i(n) = \gamma_0 < \infty \). If there exists constant \( \gamma^* > 0 \) such that the inequality \( 0 < \gamma_0 \leq \gamma^* \), then the following are true:

a) \( J(x^i(n)), i = 1, 2, \ldots, M, \) as well as \( J(y(n)) \), converge almost surely, and to the same limit.

b) \( \lim_{n \to \infty} (x^i(n) - x^j(n)) = \lim_{n \to \infty} (x^i(n) - y(n)) = 0, \forall i, j, \) almost surely and in the mean square.

c) The expression

\[
\sum_{n=1}^{\infty} \sum_{i=1}^{M} \gamma^i(n) \nabla J(x^i(n)) E[s^i(n) | F_n] \tag{3.4}
\]

is finite, almost surely. Its expectation is also finite.

Theorem 3.1 does not yet prove convergence to a minimum or a stationary point of \( J \).
Proof outline

- The difference between $y(n)$ and $x^i(n)$ for any $i$, is of the order of $A\gamma_0$, where $A$ is proportional to a bound on communication delays plus the time between consecutive communications.

- As long as $\gamma_0$ is small, $\nabla J(x^i(n))$ is approximately equal to $\nabla J(y(n))$.

- Hence $s^i(n)$ and consequently $\Phi^i(n)s^i(n)$ is approximately in a descent direction starting from $y(n)$.

- Hence iteration (2.14) is approximately the same as centralized descent algorithms which is, in general, convergent.

$$y(n + 1) = y(n) + \sum_{j=1}^{M} \gamma^j(n)\Phi^j(n)s^j(n). \quad (2.14)$$
Decreasing Step-Size Algorithms

- Let the updates $s^i(n)$ remain nonzero even if $\nabla J(x^i(n)) = 0$, that is, noise is persistent in the system.
- Situations common in stochastic approximation algorithms
- The algorithm can be made convergent only by letting the step size $\gamma^i(n)$ decreases to zero. Assume that $\gamma^i(n)$ behaves like $1/n$.
- Since the step size is decreasing, algorithm becomes progressively slower as $n \to \infty$.
- This lets the communications process to be slower as captured in Assumption 2.3

**Assumption 2.3**

There are constants $B_1 > 0$, $\beta \geq 1$, such that, for any $(j, i) \in E_l$, and for any $n$, atleast one message $x^j_l$ is sent from processor $j$ to processor $i$ during the time interval $[B_1n^\beta, B_1(n + 1)^\beta]$. Moreover the total number of messages sent and/or received during any such interval is bounded.
The next assumption allows the noise to be persistent and replaces Assumption 3.3:

**Assumption 3.4**

For some $K_1, K_2 \geq 0$ and for all $i, l, n$,

$$E[\|s_l^i(n)\|^2] \leq -K_1 E \left[ \frac{\partial J}{\partial x_l} (x_l^i(n))s_l^i(n) \right] + K_2. \quad (3.5)$$

**Theorem 3.2**

Let Assumptions 2.1, 2.3, 2.4, 3.1, 3.2 and 3.4 hold and assume that for some $K_3 \geq 0$, $\gamma_l^i(n) \leq K_3/n$, $\forall n, i$. Then Theorem 3.1 remains valid.
Corollary 3.1: Suppose that for some $K_4 > 0$, $\gamma^i(n) \geq K_4/n$, $\forall n, i$. Assume that $J$ has compact level sets and that there exist continuous functions $g^i_i: H \to [0, \infty)$ such that

$$\frac{\partial J}{\partial x^i_i}(x^i(n))E[s^i_i(n)|F_n] \leq -g^i_i(x^i(n)), \quad \forall n \in T_i^i. \quad (3.7)$$

We define $g: H \to [0, \infty)$ by $g(x) = \sum_{i=1}^M \sum_{l=1}^L g^l_i(x)$ and we assume that any point $x \in H$ satisfying $g(x) = 0$ is a stationary point of $J$. Finally, suppose that the difference between consecutive elements of $T_i^i$ is bounded, for any $i, l$ such that $T_i^i \neq \phi$. Then,

a) Under the Assumptions of either Theorem 3.1 or 3.2,

$$\liminf_{n \to \infty} \| \nabla J(x^i(n)) \| = 0, \quad \forall i, \text{ a.s.} \quad (3.8)$$

b) Under the Assumptions of Theorem 3.1 and if (for some $\epsilon > 0$) $\gamma^i(n) \geq \epsilon$, $\forall i, n$, we have

$$\lim_{n \to \infty} \| \nabla J(x^i(n)) \| = 0, \quad \forall i, \text{ a.s.} \quad (3.9)$$

and any limit point of $\{x^i(n)\}$ is a stationary point of $J$. 
Related Work

- Consensus problems are a special case where there is no cost function to optimize and the processors communicate to reach a consensus about a certain scalar.

- Recently, it has been shown that, in consensus problems, even with unbounded intercommunication delays, convergence can be guaranteed when some weak form of symmetry is present. (V.D. Blondel et al.)

- Gossip Algorithms are a further special case where the goal is the computation of the exact average on the agents’ values as opposed to reaching a consensus on some intermediate value. (S. Boyd et al.)
Thank you