A Distributed Algorithm for Finding a Common Fixed Point of a Family of Paracontractions and Some of Its Applications

A S Morse
Yale University

Lili Wang Daniel Fulmer Ji Liu

Canberra, AU December 7, 2017
Each agent has its own reception radius $r_i$.

Each agent is a neighbor of itself.

$m > 1$ agents with labels 1, 2, ..., $m$.
Motivating Problem: Solving $Ax = b$ .... over a network of $m$ agents

Each agent $i$ knows a pair of real matrices $(A_i^{n_i \times n}, b_i^{n_i \times 1})$

Standing assumption: At least one solution to the equation $Ax = b$ exists where

$$A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

There are no other assumptions about the $A_i$ and $b_i$

Time is discrete: $t \in \{1,2,...\}$

At time $t$ each agent

stores an $n \times 1$ state vector $x_i(t)$ which it can update.

receives the state vectors $x_j(t), j \in \mathcal{N}_i(t)$, of each of its current neighbors

knows nothing more

Problem: Devise local update rules for the $x_i$ which will cause all $m$ agents to iteratively arrive at the same solution to $Ax = b$
Motivating Problem

Algorithm

Positive definite gain matrix chosen small enough so that

\[ \text{spectrum} \left( I - A_i^0 G_i A_i \right)^{1/2} (1; 1) \]

\[
M_i(x) = x - A_i' G_i (A_i x - b_i)
\]

\[
x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\}
\]

Number of labels in \( \mathcal{N}_i(t) \)

\[
A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
\]

\[\begin{align*}
Ax &= b &\iff& A_i x = b_i, \quad i \in \{1, 2, \ldots, m\} \\
A_i x &= b_i &\iff& M_i(x) = x
\end{align*}\]

Therefore any solution to \( Ax = b \) is a common fixed point of the \( M_i \) and conversely.

Can a distributed algorithm be constructed for computing a common fixed point for other types of possibly nonlinear \( M_i \)?
Problem Generalization

Each agent $i$ knows a (possibly nonlinear) map

$$M_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

Suppose the $M_i$ have at least one common fixed point $y$; that is

$$M_i(y) = y, \quad i \in \{1, 2, \ldots, m\}$$

Each agent $i$ generates a state $x_i(t) \in \mathbb{R}^n$ which is its current estimate of $y$.

Each agent $i$ receives the state $x_j(t)$ of each of its current neighbors $j \in \mathcal{N}_i(t)$.

**Problem:** Devise a distributed recursive algorithm which enables all $m$ agents to asymptotically compute the same common fixed point of the $M_i$.

But for what kinds of $M_i$?

$$M_i(x) = x - A_i'G_i(A_ix - b_i) \quad \sigma(I - A_i'G_iA_i) \subset (-1, 1)$$

Each $M_i$ is a “paracontraction” with respect to the 2-norm.
Paracontraction

\( M : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is a paracontraction with respect to a given norm \( \| \cdot \| \) on \( \mathbb{R}^n \) if it is continuous and

\[
\| M(x) - y \| < \| x - y \|
\]

for all \( y \in \mathbb{R}^n \) satisfying \( M(y) = y \) and all \( x \in \mathbb{R}^n \) satisfying \( M(x) \neq x \)

Thus if \( M \) is a paracontraction and \( x \) is not a fixed point, then \( M(x) \) is “closer” to the set of fixed points

\[
\mathcal{F}(M) = \{ z : M(z) = z \}
\]

than \( x \) was.
Paracontraction \( w/r \| \cdot \|_2 \)

**Example**

**Orthogonal Projection**

\[ M(x) = \arg \min_{y \in \mathcal{C}} \|x - y\|_2 \]

\( \mathcal{C} = \text{closed convex set} \)

Fixed points of \( M \) are vectors in \( \mathcal{C} \)

Can be used to find a vector in the intersection of \( m \) closed convex sets.
The Problem {again}

Paracontractions: \[ M_i : \mathbb{R}^n \to \mathbb{R}^n, \quad i \in \{1, 2, \ldots, m\} \]

Common fixed point \( y \): \[ M_i(y) = y, \quad i \in \{1, 2, \ldots, m\} \]

Agent \( i \) has a state \( x_i(t) \) which is to be its current estimate of a common fixed point.

Agent \( i \) knows \( M_i \) and receives the state \( x_j(t) \) of each of its current neighbors \( j \in \mathcal{N}_i(t) \).

**Problem:** Devise a distributed recursive algorithm which enables all \( m \) agents to asymptotically compute the same common fixed point of the \( M_i \).
The Algorithm

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

Generalization:

\[ x_i(t + 1) = M_i \left( \sum_{j \in \mathcal{N}_i(t)} w_{ij}(t) x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

\[ w_{ij}(t) \geq 0 \quad \sum_{j \in \mathcal{N}_i(t)} w_{ij}(t) = 1 \]

Require that for each \( i \) and \( j \), and all \( t \geq 1 \), \( w_{ij}(t) \in \mathcal{W}_{ij} = \text{finite set} \)
Neighbor Graph

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

\( \mathcal{G} = \) all directed graphs with vertex set \( \mathcal{V} = \{1, 2, \ldots, m\} \)

\( \mathcal{N}(t) = \) graph in \( \mathcal{G} \) with an arc from \( j \) to \( i \) whenever \( j \in \mathcal{N}_i(t), \quad i \in \{1, 2, \ldots, m\} \)

\( j \) is a neighbor of \( i \)
Main Result

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

Suppose that for some finite integer \( p > 1 \), the maps \( M_1, M_2, \ldots, M_m \) are all paracontractions with respect to the same norm \( \| \cdot \|_p \) and, in addition, that the \( M_i \) all share at least one common fixed point. Then for any sequence of strongly connected neighbor graph \( \mathbb{N}(t), \ t \geq 1 \), all \( x_i(t) \) converge to the same point as \( t \to \infty \), and this point is a fixed point of all of the \( M_i \).

\{NOLCOS 2016\}

This result also holds if the sequence of neighbor graphs \( \mathbb{N}(t), \ t \geq 1 \), is “repeatedly jointly strongly connected.”

Repeatedly jointly strongly connected means that for some finite integer \( T > 0 \), the union of each successive sub-sequence of \( T \) graphs in the sequence \( \mathbb{N}(t), \ t \geq 1 \), is strongly connected.

This result also holds if the agents do not share a common clock and updates are performed asynchronously. \{CDC 2016\}
Elsner, Koltracht, Neumann (1992):

Suppose $\mathcal{P}$ is a finite set of paracontractions $P : \mathbb{R}^n \to \mathbb{R}^n$ with respect to some norm on $\mathbb{R}^n$. Suppose the paracontractions share a common fixed point. Then the state of the iteration

$$z(t + 1) = P_t(z(t)), \quad t \geq 1$$

converges to a common fixed point of the paracontractions which occur in the sequence $P_t$, $t \geq 1$, infinitely often.
\[ M_i(x) = x - A_i'G_i(A_ix - b_i) \]

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in N_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

Assume for simplicity that each \( A_i \) has linearly independent rows

Simulation

Positive definite gain matrix chosen small enough so that

\[ \text{specturm} \left( I_i A_i^0 G_i A_i \right)^{1/2} \left( I; 1^1 \right) \]
Simulation

\[ M_i(x) = x - A'_i G_i (A_i x - b_i) \]

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in N_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

Assume for simplicity that each \( A_i \) has linearly independent rows
Simulation

\[ M_i(x) = x - A_i' G_i (A_i x - b_i) \]

\[ (A_i A_i')^{-1} \leq G_i < 2(A_i A_i')^{-1} \]

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

Assume for simplicity that each \( A_i \) has linearly independent rows
\[ y - x = 1 \]
\[ y + 3x = 3 \]

\[ A_1 = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad b_1 = 1 \]
\[ A_2 = \begin{bmatrix} 3 & 1 \end{bmatrix} \quad b_2 = 3 \]

\[ M_i(x) = x - A'_i G_i (A_i x - b_i) \quad (A_i A'_i)^{-1} \leq G_i < 2 (A_i A'_i)^{-1} \]

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

\[ G_i = (A_i A'_i)^{-1}, \quad i = 1, 2 \]
\[ G_1 = \frac{2}{3} I \quad G_2 = \frac{1}{6} I \]

Two simulations: one with

the other with
\[ y - x = 1 \quad A_1 = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad b_1 = 1 \]
\[ y + 3x = 3 \quad A_2 = \begin{bmatrix} 3 & 1 \end{bmatrix} \quad b_2 = 3 \]

\[ M_i(x) = x - A_i'G_i(A_ix - b_i) \quad (A_iA_i')^{-1} \leq G_i < 2(A_iA_i')^{-1} \]

\[ x_i(t + 1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in \mathcal{N}_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

\[ G_i = (A_iA_i')^{-1}, \quad i = 1, 2 \]

\[ G_1 = \frac{2}{3}I = \frac{1}{6}I \]
\[ \begin{align*}
    y - x &= 1 \\
    y + 3x &= 3
\end{align*} \]

\[ A_1 = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad b_1 = 1 \]

\[ A_2 = \begin{bmatrix} 3 & 1 \end{bmatrix} \quad b_2 = 3 \]

\[ M_i(x) = x - A_i' G_i( A_i x - b_i ) \]

\[ (A_i A_i')^{-1} \leq G_i < 2(A_i A_i')^{-1} \]

\[ x_i(t+1) = M_i \left( \frac{1}{m_i(t)} \sum_{j \in N_i(t)} x_j(t) \right), \quad t \geq 1, \quad i \in \{1, 2, \ldots, m\} \]

\[ G_i = (A_i A_i')^{-1}, \quad i = 1, 2 \]

\[ G_1 = \frac{2}{3} I \quad G_2 = \frac{1}{6} I \]
Application: Distributed State Estimation
Joint observability: \( ( \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_m \end{bmatrix}, A ) \) is an observable pair

\[ C_i \neq 0 \quad i \in \{1, 2, \ldots, m\} \]

Objective: Devise a family of \( m \) estimators, one for each agent, whose outputs \( x_i, \ i \in \{1,2,\ldots,m\} \) are all asymptotically correct estimates of \( x \)

Suppose that each agent \( i \) generates estimate \( x_i \) as the output of an \( n_i \) - dimensional linear time invariant system with state \( z_i \)
Objective: Devise a family of $m$ estimators, one for each agent, whose outputs $x_i$, $i \in \{1, 2, \ldots, m\}$ are all asymptotically correct estimates of $x$.

Suppose that each agent $i$ generates estimate $x_i$ as the output of an $n_i$-dimensional linear time invariant system with state $z_i$.

Suppose that each agent $i$ broadcasts its state $z_i$ and measurement $y_i$.
First construct a “classical observer” for each agent $i$ which is capable of correctly estimating that “part of the process state,” i.e., $L_i x$, which is observable to agent $i$:

Here $L_i$ is any $n_i \times n$ matrix for which

$$
\ker L_i = \ker \begin{bmatrix} C_i \\ C_i A \\ \vdots \\ C_i A^{n-1} \end{bmatrix} 
$$

and note that $(\bar{C}_i, \bar{A}_i)$ is an observable pair.

Choose $K_i$ to make the convergence rate of $\bar{A}_i + K_i \bar{C}_i$ as large as desired.

Local observer $i$:  
$$
\dot{w}_i = (\bar{A} + K_i \bar{C}_i) w_i + y_i
$$

Estimation error:  
$$
\epsilon_i = w_i - L_i x \quad \text{satisfies} \quad \epsilon_i = e^{(\bar{A}_i + K_i \bar{C}_i) t} \epsilon_i(0)
$$
\[ w_i(t) = L_i x(t) + \epsilon_i(t) \quad i \in \{1, 2, \ldots, m\} \]

Estimation error: \[ \epsilon_i = w_i - L_i x \] satisfies \[ \epsilon_i = e^{(A_i + K_i C_i) t} \epsilon_i(0) \]
\[ w_i(t) = L_i x(t) + \epsilon_i(t) \quad i \in \{1, 2, \ldots, m\} \]

Pick a set of equally spaced “event times” \( t_0, t_1, t_2, \ldots \ldots \) where \( t_0 = 0 \) and \( t_j - t_{j-1} = T, j \geq 1 \).

Generate agent \( i \)'s estimate \( x_i(t) \) of \( x(t) \) as follows:

1. For each fixed \( j > 0 \), iterate the previously discussed linear equation solver \( q \) times within the interval \([t_{j-1}, t_j]\) to obtain an estimate \( z_{ij} \) of the parameter \( p_j \) assuming \( p_j \) satisfies the equations

\[ w_i(t_{j-1}) = L_i p_j, \quad i \in \{1, 2, \ldots, m\} \]

2. Prompted by the fact that

\[ w_i(t_{j-1}) = L_i x(t_{j-1}) + \epsilon_i(t_{j-1}) \quad i \in \{1, 2, \ldots, m\} \]

take \( z_{ij} \) to be an after the fact estimate of \( x(t_{j-1}) \) and define

\[ x_i(t_j) = e^{AT} z_{ij} \]

3. Between event times generate \( x_i(t) \) using

\[ \dot{x}_i = Ax_i \]

Can prove that with \( q \) properly chosen and with appropriate network connectivity (e.g., strong connectivity \( \forall t \)), all \( x_i(t) \) converge to \( x(t) \) exponentially fast at any pre-specified rate. CDC 2017
Additional Properties

**Robustness:** The estimation is robust to small differences in agents' event times provided $A$ is a stability matrix.

**Asynchronism:** Linear equation solver computations can be performed asynchronously.

**Resilience:** With enough redundancy the overall estimator is "resilient."
Resilience

By a *resilient algorithm* for a distributed process is meant an algorithm which, by exploiting built-in network and data redundancies, is able to continue to function correctly in the face of abrupt changes in the number of nodes and edges in the inter-agent communication graph upon which the algorithm depends.

Such changes might arise as a result of a network communication failure, a component failure, a sensor temporarily being put to sleep to conserve energy, or even possibly a malicious attack.

Distributed estimators which are *linear time-invariant* systems are not resilient.

It is easy to see that the hybrid estimator just described, is.
Simulations

\[ \mathbf{x} = A \mathbf{x} + \mathbf{b} \quad \mathbf{v} = \text{noise} \]

\[ y_i = C_i \mathbf{x}; \quad i \in \{1, 2, 3, 4\} \]

System: 4 agent, stable, 4-dimensional with eigenvalues at -0.1, -0.1, -0.05 ± \( j0.614 \)
Simulations

\[ \begin{align*}
\dot{x} &= Ax + bv \\
y_i &= C_i x; \quad i \in \{1, 2, 3, 4\}
\end{align*} \]

System: 4 agent, stable, 4-dimensional with eigenvalues at -0.1, -0.1, -0.05 ± j0.614

redundantly jointly observable – remains jointly observable if one agent dies.

Neighbor graph:

Redundant strongly connected
Remains strongly connected if any one vertex is removed.
Not a complete graph

Simulation with \( \nu \) a sinusoid \( = 7\cos(10t) \)
Simulation with \( \nu \) white noise \( = \{0 \text{ mean, variance } 1\} \)

In both simulations, agent 4 leaves the network at \( t = 5 \) and returns at \( t = 7 \)
Simulations

\[ \nu = \text{sine wave} \]

Agent 4 leaves the network at \( t = 5 \) and returns at \( t = 7 \)

\[ x = Ax + bv \]

\[ y_i = C_i x; \ i \in \{1, 2, 3, 4\} \]

3rd components of \( x \) and \( x_1 \)

State of estimator 1

\[ \| x_1 - x \|_2 \]
Simulations

$\mathbf{\nu} =$ white noise

$x_1 = A x + b \mathbf{\nu}$

$y_i = C_i x; i \in \{1, 2, 3, 4\}$

Agent 4 leaves the network at $t = 5$ and returns at $t = 7$.
Why is Stability Needed for Robustness?

Process: \[ y = Cx \quad \dot{x} = Ax \quad A = \text{unstable} \]

Estimator: \[ \dot{x} = \hat{A}x + K(C\hat{x} - y) \quad \hat{A} - A = \text{small} \quad \hat{A} + KC = \text{stable} \]

Estimation Error: \[ e = \hat{x} - x \]

\[ \dot{e} = (\hat{A} + KC)e + (\hat{A} - A)x \]

Therefore for state estimation problems for processes without controlled inputs, internal stability must be assumed if the estimation problem is to make sense!

The same conclusion applies to distributed state estimation problems.
Concerning para-contractions, it would be interesting to

1. derive necessary conditions on neighbor graph sequences for convergence
2. derive worst case bounds on convergence rate
3. discover other types of maps which are paracontractions

Concerning hybrid estimators, it would be interesting to

1. study what happens if different agents have different clocks
2. take into account transmission delays across the network

Concerning linear time invariant distributed observers it would be interesting to

1. prove that they cannot be resilient