Finite-Time Distributed Consensus in Graphs with Time-Invariant Topologies

Shreyas Sundaram and Christoforos N. Hadjicostis

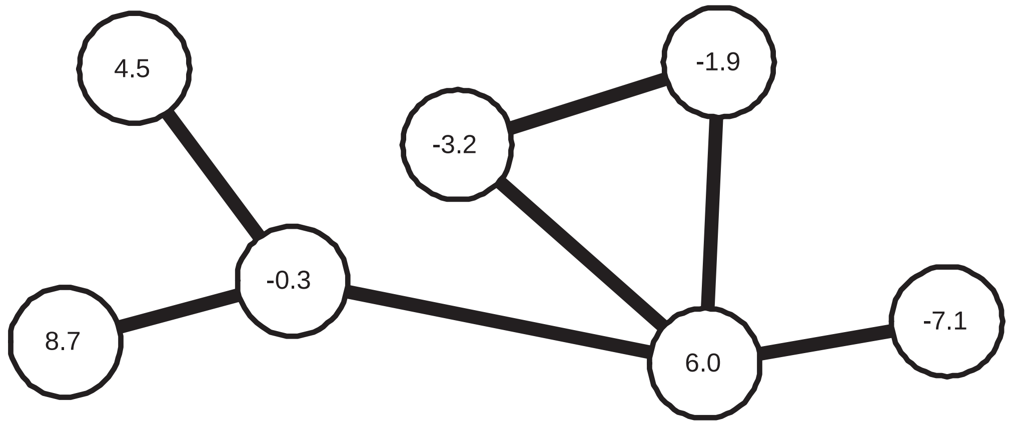
Coordinated Science Laboratory

and

Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

Overview



- Consider a network (directed or undirected) with nodes $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$
- Each node can only receive information from its neighbors
- Each node i has some initial value $x_i[0]$
  - e.g., temperature measurement, position, vote, fault status, etc.
- **Consensus:** All nodes must calculate the same function of the initial values
  - e.g., average, max, min, mode, etc.
  - Studied for several decades (e.g., [Garg, *Elements of Distributed Computing*])

Linear Iterations for Consensus

- Here, we focus on linear iteration-based schemes
  - At each time-step, each node i updates its value to be a linear combination of itself and its neighbors:

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in \text{nbrs}_i} w_{ij}x_j[k]$$

  - Update equation for entire system:

$$\underbrace{\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \\ \vdots \\ x_N[k+1] \end{bmatrix}}_{x[k+1]} = \underbrace{\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_N[k] \end{bmatrix}}_{x[k]}$$

    - Constraint: $w_{ij} = 0$ if x_j is not a neighbor of x_i
- **Asymptotic Consensus:** Choose W so that $\lim_{k\to\infty} x[k] = \mathbf{1}\mathbf{c}'x[0]$
  - $\mathbf{1}$ is column vector with all 1's, and $\mathbf{c}'$ is some row vector

Convergence in Time-Invariant Graphs

- Linear iteration schemes well studied in control systems literature (e.g., survey papers [Ren, Beard, & Atkins, ACC'05], [Olfati-Saber, Fax, & Murray, *Proc. IEEE*, 2007])
- Necessary and sufficient conditions for $x[k] \rightarrow \mathbf{1}\mathbf{c}'x[0]$ [Xiao & Boyd, 2004]:
  - $W\mathbf{1} = \mathbf{1}$, $\mathbf{c}'W = \mathbf{c}'$
  - All other eigenvalues of W must have magnitude strictly less than 1
- Rate of convergence given by second largest eigenvalue of W
  - Faster convergence $\rightarrow$ minimize second largest eigenvalue by choosing weights appropriately [Xiao & Boyd, 2004]
- Almost all existing methods only consider **asymptotic** convergence

Contribution: We show how to obtain **finite-time** consensus via linear iterations

Finite-Time Consensus via Linear Iterations

- Let W be any weight matrix providing asymptotic consensus (for now)
  - For simplicity, assume W is symmetric
- Denote distinct eigenvalues of W by $\lambda_1, \lambda_2, \ldots, \lambda_D$
- **Minimal polynomial** of W:

$$\begin{aligned} q(t) &= (t - \lambda_1)(t - \lambda_2) \cdots (t - \lambda_D) \\ &\equiv t^D + \alpha_{D-1} t^{D-1} + \cdots + \alpha_1 t + \alpha_0 \end{aligned}$$

- Minimal polynomial satisfies $q(W) = 0$:

$$W^D + \alpha_{D-1} W^{D-1} + \cdots + \alpha_1 W + \alpha_0 I = 0$$

- Using $x[k+r] = W^r x[k]$:

$$\begin{aligned} & x[k+D] + \alpha_{D-1} x[k+D-1] + \cdots + \alpha_1 x[k+1] + \alpha_0 x[k] = 0 \\ \Rightarrow \quad & x_i[k+D] + \alpha_{D-1} x_i[k+D-1] + \cdots + \alpha_1 x_i[k+1] + \alpha_0 x_i[k] = 0 \end{aligned}$$

Closed Form Expression for Consensus Value (1)

- Linear difference equation:

$$x_i[k+D] + \alpha_{D-1} x_i[k+D-1] + \cdots + \alpha_1 x_i[k+1] + \alpha_0 x_i[k] = 0$$

- Take z-transform of above expression:

$$X_i(z) = \frac{\sum_{j=0}^{D-1} x_i[j] z^{D-j} + \alpha_{D-1} \sum_{j=0}^{D-2} x_i[j] z^{D-1-j} + \cdots + \alpha_1 z x_i[0]}{\underbrace{(z^D + \alpha_{D-1} z^{D-1} + \cdots + \alpha_1 z + \alpha_0)}_{q(z)}}$$

- Roots of $q(z)$ are eigenvalues of W
- W has a simple eigenvalue at 1, and all other eigenvalues inside unit circle
  - Poles of $(z-1)X_i(z)$ are stable
- Final value theorem:

$$\lim_{k\to\infty} x_i[k] = \lim_{z\to 1} (z-1) X_i(z)$$

Closed Form Expression for Consensus Value (2)

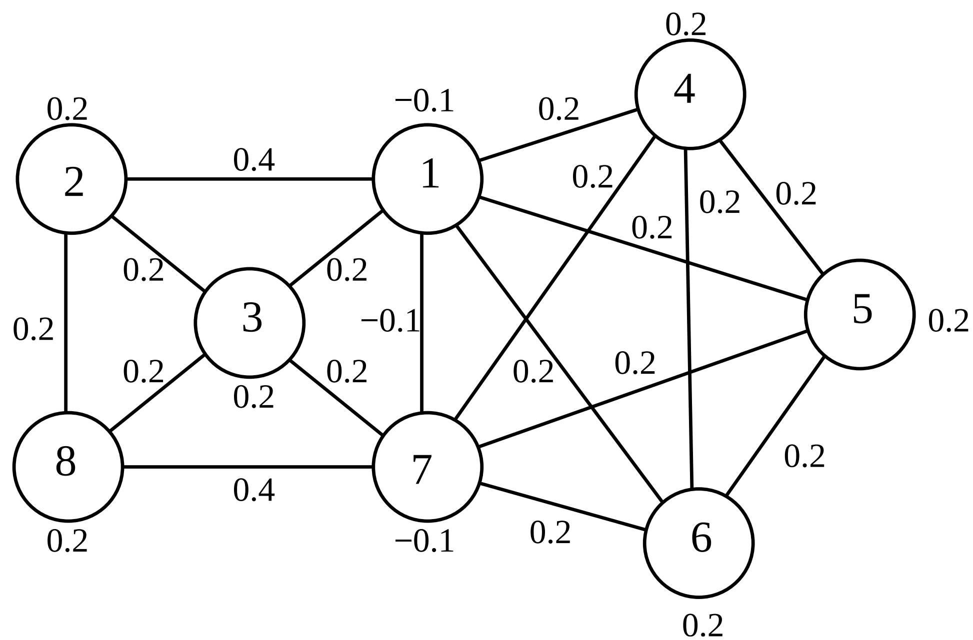
- Consensus value for node i, $1 \leq i \leq N$:

$$\lim_{k\to\infty} x_i[k] = \frac{\begin{bmatrix} x_i[D-1] & x_i[D-2] & \cdots & x_i[0] \end{bmatrix} \begin{bmatrix} 1 \\ 1+\alpha_{D-1} \\ 1+\alpha_{D-2}+\alpha_{D-1} \\ \vdots \\ 1+\sum_{j=1}^{D-1}\alpha_j \end{bmatrix}}{\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1+\alpha_{D-1} \\ 1+\alpha_{D-2}+\alpha_{D-1} \\ \vdots \\ 1+\sum_{j=1}^{D-1}\alpha_j \end{bmatrix}}$$

Key Result:

Each node can calculate the consensus value after D time-steps as a linear combination of its own values over those time-steps

Example: The Network and Weights



- Consider example from [Xiao & Boyd, 2004]
- Weights on edges and nodes chosen to maximize asymptotic rate of convergence
- W has simple eigenvalue 1, with $W\mathbf{1} = \mathbf{1}$, $\frac{1}{8}\mathbf{1}'W = \frac{1}{8}\mathbf{1}'$

Example: Finding the Coefficient Vector

- Eigenvalues of W are $\{1, 0.6, 0.4, 0, 0, 0, -0.4, -0.6\}$
- Minimal polynomial of W is

$$\begin{aligned} q(t) &= (t-1)(t-0.6)(t-0.4)t(t+0.4)(t+0.6) \\ &= t^6 - t^5 - 0.52t^4 + 0.52t^3 + 0.0576t^2 - 0.0576t \\ &= t^6 + \alpha_5 t^5 + \alpha_4 t^4 + \alpha_3 t^3 + \alpha_2 t^2 + \alpha_1 t + \alpha_0 \end{aligned}$$

- Minimal polynomial has degree 6, so nodes can reach consensus in 6 time-steps
- Define vector

$$S = \begin{bmatrix} 1 \\ 1+\alpha_5 \\ 1+\alpha_4+\alpha_5 \\ 1+\alpha_3+\alpha_4+\alpha_5 \\ 1+\alpha_2+\alpha_3+\alpha_4+\alpha_5 \\ 1+\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -0.52 \\ 0 \\ 0.0576 \\ 0 \end{bmatrix}$$

Example: Running the Linear Iteration

- Suppose initial values of the nodes are

$$x[0] = \begin{bmatrix} 1.3889 & 2.0277 & 1.9872 & 6.0379 & 2.7219 & 1.9881 & 0.1527 & 7.4679 \end{bmatrix}'$$

- Run iteration $x[k+1] = Wx[k]$ for 6 time-steps:

$$\begin{bmatrix} x[5] & x[4] & x[3] & x[2] & x[1] & x[0] \end{bmatrix} =$$

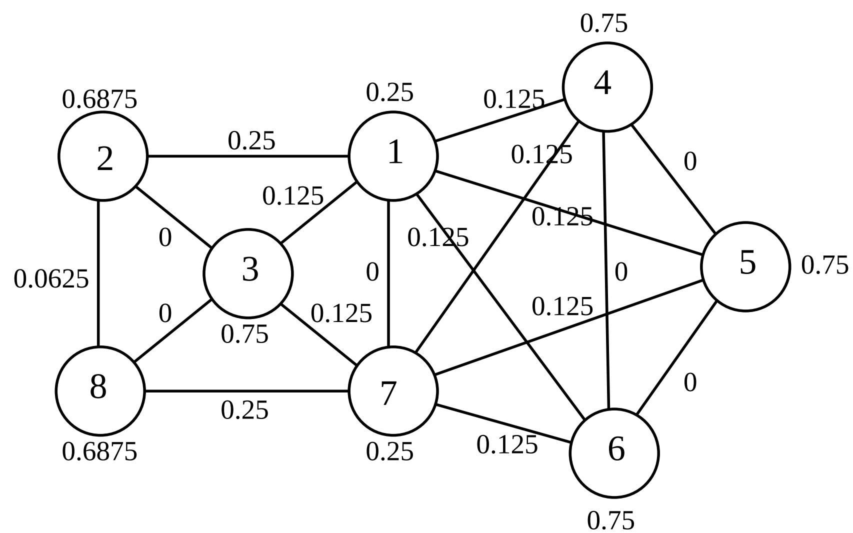
$$\begin{bmatrix} 3.1148 & 2.7022 & 3.2728 & 2.2782 & 3.2039 & 1.3889 \\ 2.9303 & 3.0128 & 2.8791 & 2.8445 & 2.8521 & 2.0277 \\ 2.9240 & 3.0825 & 2.8395 & 3.2797 & 2.6049 & 1.9872 \\ 2.9050 & 3.0507 & 2.7866 & 3.1915 & 2.4579 & 6.0379 \\ 2.9050 & 3.0507 & 2.7866 & 3.1915 & 2.4579 & 2.7219 \\ 2.9050 & 3.0507 & 2.7866 & 3.1915 & 2.4579 & 1.9881 \\ 3.1705 & 2.6705 & 3.6210 & 2.0804 & 5.3800 & 0.1527 \\ 2.9177 & 3.1521 & 2.8000 & 3.7149 & 2.3576 & 7.4679 \end{bmatrix}$$

Example: Calculating the Consensus Value

- Each node i calculates consensus value as

$$\frac{\begin{bmatrix} x_i[5] & x_i[4] & x_i[3] & x_i[2] & x_i[1] & x_i[0] \end{bmatrix} S}{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} S} = 2.9715$$

Question: Do optimal asymptotic weights produce minimal-time consensus?



- Above weights produce W with eigenvalues $\{1, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{8}, 0\}$
- These weights provide finite-time consensus in 4 time-steps, but are worse for asymptotic consensus!

More General Weight Matrices

Question: Can we use more general weight matrices to obtain finite-time consensus?

Theorem:

- Suppose W has a simple eigenvalue μ, with

$$W\mathbf{d} = \mu\mathbf{d}, \;\; \mathbf{c}'W = \mu\mathbf{c}'$$

  where $\mathbf{d}$ has all entries nonzero
- Let D denote degree of minimal polynomial of W
- Then there exists a set of coefficients $\gamma_0, \gamma_1, \ldots, \gamma_{D-1}$ for each node i such that

$$\mathbf{c}'x[0] = \gamma_{D-1}x_i[D-1] + \gamma_{D-2}x_i[D-2] + \cdots + \gamma_0 x_i[0]$$

  - These coefficients are obtained from coefficients of minimal polynomial

Note: Magnitude of eigenvalues does not matter!

Choosing the Weight Matrix

- Conditions are satisfied by any W providing asymptotic convergence
  - Various methods of choosing such weights in literature
- However, other interesting choices also exist, such as

$$w_{ij} = \begin{cases} 1, & \text{if } x_j \text{ is a neighbor of } x_i \\ 0, & \text{if } x_j \text{ is not a neighbor of } x_i \\ N - \sum_{i \neq j} w_{ij}, & \text{if } i = j. \end{cases}$$

  - This choice produces $\mu = N$, $\mathbf{d} = \mathbf{1}$, $\mathbf{c}' = \frac{1}{N}\mathbf{1}'$
  - W consists of only integers, possibly allowing easier computations

Finding the Minimal Polynomial: Decentralized Method

- Nodes require minimal polynomial $q(t)$ to calculate coefficients for finite-time consensus
- Can the nodes calculate the minimal polynomial in a decentralized manner?
- Suppose $q(t) = t^D + \alpha_{D-1}t^{D-1} + \cdots + \alpha_0$
- We saw earlier that

$$
\begin{aligned}
0 &= x_i[D] + \alpha_{D-1}x_i[D-1] + \cdots + \alpha_0 x_i[0] \\
&= \begin{bmatrix} x_i[D] & x_i[D-1] & x_i[D-2] & \cdots & x_i[1] & x_i[0] \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_{D-1} \\ \vdots \\ \alpha_0 \end{bmatrix}
\end{aligned}
$$

Idea: Nodes run several linear iterations and find coefficients of minimal polynomial from null space of their values

Finding the Minimal Polynomial: Decentralized Method

- Nodes perform N runs of linear iteration with different sets of initial conditions
- Let $x_{i,j}[k]$ denote node i's value at time-step k of the j'th run
- Node i finds smallest D for which

$$\begin{bmatrix} x_{i,1}[D] & x_{i,1}[D-1] & \cdots & x_{i,1}[0] \\ x_{i,2}[D] & x_{i,2}[D-1] & \cdots & x_{i,2}[0] \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,N}[D] & x_{i,N}[D-1] & \cdots & x_{i,N}[0] \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_{D-1} \\ \vdots \\ \alpha_0 \end{bmatrix} = 0$$

  has nontrivial solution

- If the N initial conditions are linearly independent, coefficients of $q(t)$ are given by above equation
- Node i can now obtain coefficients for finite time consensus

Summary

- Can obtain distributed consensus after running linear iterations for a finite number of time-steps
- Minimal polynomial of weight matrix provides coefficients for consensus value
- Nodes can learn minimal polynomial in a decentralized manner
- Eigenvalues of weight matrix can have arbitrary magnitudes

Extensions Since September and Future Plans

Building on these results and using concepts from observability theory, we have obtained the following extensions:

- **Arbitrary consensus functions:** Nodes can reach consensus on any arbitrary function of initial values in finite time
- **Function calculation:** Different nodes can calculate different functions of initial values
- **Noisy consensus:** Nodes can calculate bounded variance estimates of consensus value in the presence of communication and update noise
- **Fault tolerance:** Despite nodes that update their values erroneously or maliciously, other nodes can still reach consensus on the desired function

Open problems:

- minimal-time consensus, time-varying graphs, quantized consensus via linear iterations, trust assessment, . . .