Lecture #1: Introduction to distributed algorithms

Francesco Bullo[1] Jorge Cortés[2] Sonia Martínez[2]

[1]Department of Mechanical Engineering
University of California, Santa Barbara
bullo@engineering.ucsb.edu

[2]Mechanical and Aerospace Engineering
University of California, San Diego
{cortes,soniamd}@ucsd.edu

Workshop on "Distributed Control of Robotic Networks"
IEEE Conference on Decision and Control
Cancun, December 8, 2008

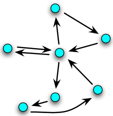
Content Summary

A motivating example

Simplest distributed iteration is **linear averaging**:

- you are given a graph
- each node contains a value x_i
- each node repeatedly executes:

$$x_i^+ := \mathsf{average}(x_i, \{x_j, \text{for all neighboring } j\})$$



Why does this algorithm converge and to what?

Matrix theory: matrix sets

A matrix $A \in \mathbb{R}^{n\times n}$ with entries a_{ij}, $i, j \in \{1, \dots, n\}$, is

1. **nonnegative** (resp., **positive**) if all its entries are nonnegative (resp., positive)
2. **row-stochastic** (or **stochastic** for brevity) if it is nonnegative and $\sum_{j=1}^n a_{ij} = 1$, for all $i \in \{1, \dots, n\}$; that is

$$A\mathbf{1}_n = \mathbf{1}_n$$

3. **doubly stochastic** if it is row-stochastic and column-stochastic
4. a **permutation matrix** if A has precisely one entry equal to 1 in each row, one entry equal to 1 in each column, and all other entries equal to 0 (note: every permutation is doubly stochastic)

Matrix sets: properties

- row-stochastic matrix: each row is a "convex combination"
- row-stochastic matrix: $A\mathbf{1}_n = \mathbf{1}_n$ means 1 is eigenvalue
- column-stochastic map preserves "vector sum"

$$v \mapsto Av, \qquad \sum_{i=1}^{n} (Av)_i = \mathbf{1}_n^T A v = \mathbf{1}_n^T v = \sum_{i=1}^{n} v_i$$

Birkhoff–Von Neumann Theorem

Equivalent statements:

- A is doubly stochastic
- A it is a convex combination of permutation matrices

Matrix sets: cont'd

A non-negative matrix $A \in \mathbb{R}^{n \times n}$ with entries a_{ij}, $i, j \in \{1, \dots, n\}$, is

1. **irreducible** if, for any nontrivial partition $J \cup K$ of the index set $\{1, \dots, n\}$, there exists $j \in J$ and $k \in K$ such that $a_{jk} \neq 0$
   or, is **reducible** if there exists a permutation matrix P such that $P^T A P$ is block upper triangular
2. **primitive** if there exists $k \in \mathbb{N}$ such that A^k is positive

(primitive implies irreducible)

Bad examples: A_1 reducible and A_2 irreducible, but not primitive:

$$A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Good examples: Non-negative, irreducible, and primitive:

$$A_3 = \frac{1}{2}\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad A_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Convergent matrices

Convergent and semi-convergent matrices

A square matrix A is

1. **convergent** if $\lim_{\ell \to +\infty} A^\ell$ exists and $\lim_{\ell \to +\infty} A^\ell = 0$
2. **semi-convergent** if $\lim_{\ell \to +\infty} A^\ell$ exists

Spectral radiuses

Given a square matrix A,

- its **spectral radius** is

$$\rho(A) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(A)\}$$

- if $\rho(A) = 1$ (e.g., A stochastic), then **essential spectral radius**

$$\rho_{\text{ess}}(A) = \max\{\|\lambda\|_{\mathbb{C}} \mid \lambda \in \operatorname{spec}(A) \setminus \{1\}\}$$

Convergent matrices: cont'd

Necessary and sufficient conditions for convergence

A is convergent if and only if $\rho(A) < 1$

Recall: row-stochastic matrix has eigenvalue 1
Indeed, row-stochastic matrix has spectral radius 1

Necessary and sufficient conditions for semi-convergence

A is semi-convergent if and only if

1. $\rho(A) \leq 1$
2. $\rho_{\text{ess}}(A) < 1$
   i.e., 1 is an eigenvalue and is the only eigenvalue on the unit circle
3. the eigenvalue 1 is **semisimple**
   i.e., 1 has equal algebraic and geometric multiplicity ≥ 1

Perron-Frobenius theory

Perron-Frobenius theorem

Assume A is **positive**, or
assume A is **non-negative, irreducible and primitive**, then

1. $\rho(A) > 0$
2. $\rho(A)$ is an **eigenvalue** that is **simple** and **strictly larger** than the magnitude of any other eigenvalue
3. $\rho(A)$ has an eigenvector with positive components

Implication for stochastic matrices

A is stochastic, irreducible and primitive $\implies$ A is semiconvergent

Implication for linear averaging

Graph is such that A is primitive $\implies$ linear averaging algorithm is convergent

Basic graph notions

A **directed graph** or **digraph**, of order n is $G = (V, E)$

- V is set with n elements – **vertices**
- E is set of ordered pair of vertices – **edges**

Digraph is **complete** if $E = V \times V$. (u, v) denotes an edge **from** u **to** v

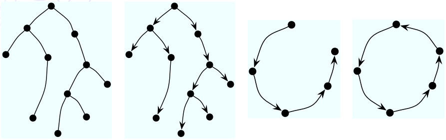
An **undirected graph** consists of a vertex set V and of a set E of unordered pairs of vertices. $\{u, v\}$ denotes an unordered edge

A digraph (V', E') is

- **undirected** if $(v, u) \in E'$ anytime $(u, v) \in E'$
- a **subgraph** of a digraph (V, E) if $V' \subset V$ and $E' \subset E$
- a **spanning subgraph** if it is a subgraph and $V' = V$

Example graphs

Tree, directed tree, chain, and ring digraphs:



Graph neighbors

In a digraph G with an edge $(u, v) \in E$, u is **in-neighbor** of v, and v is **out-neighbor** of u

$\mathcal{N}_G^{\text{in}}(v)$: set of in-neighbors of v – cardinality is **in-degree**
$\mathcal{N}_G^{\text{out}}(v)$: set of out-neighbors of v – cardinality is **out-degree**

A digraph is **topologically balanced** if each vertex has the same in- and out-degrees, i.e., same number of incoming and outgoing edges

Likewise, u and v are **neighbors** in a graph G if $\{u, v\}$ is an undirected edge

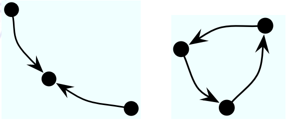
$\mathcal{N}_G(v)$: set of neighbors of v in the undirected graph G – cardinality is **degree**

Connectivity notions

- A **directed path** in a digraph is an ordered sequence of vertices such that any ordered pair of vertices appearing consecutively in the sequence is an edge of the digraph
- A vertex of a digraph is **globally reachable** if it can be reached from any other vertex by traversing a directed path.
- A digraph is **strongly connected** if every vertex is globally reachable
- A **directed tree** is a digraph such that

  there exists a vertex, called ***root****, such that any other vertex of the digraph can be reached by one and only one path starting at the root*
- In a directed tree, every in-neighbor is a **parent** and every out-neighbor is a **child**.
- **Directed spanning tree** = spanning subgraph + directed tree

Connectivity notions: cont'd



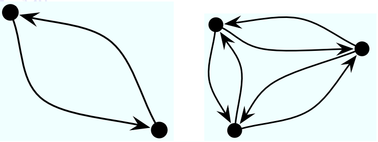
1. digraph with one sink and two sources
2. directed path which is also a cycle

Cycles and periodicity

Given a digraph G

- a **cycle** is a non-trivial directed path that
  1. starts and ends at the same vertex
  2. contains no repeated vertex except for initial and final
- G is **acyclic** if it contains no cycles
- G contains a finite number of cycles
- G is **aperiodic** if there exists no $k > 1$ that divides the length of every cycle of the graph.
- i.e., G aperiodic if the greatest common divisor of cycle lengths is 1

Cycles and periodicity: cont'd

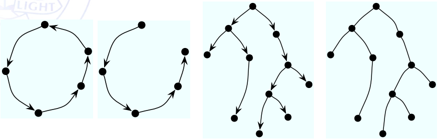


(a) (b)

Figure: (a) A digraph whose only cycle has length 2 is periodic. (b) A digraph with cycles of length 2 and 3 is aperiodic.

Example graphs

Ring digraph, chain digraph (also called path digraph), directed tree, tree



Connectivity in topologically balanced digraphs

Connectivity characterizations

Let G be a digraph:

1. G is strongly connected $\implies$ G contains a globally reachable vertex and a spanning tree
2. G is topologically balanced and contains either a globally reachable vertex or a spanning tree $\implies$ G is strongly connected

Analogous definitions can be given for the case of undirected graphs. If a vertex of a graph is globally reachable, then every vertex is, the graph contains a spanning tree, and we call the graph **connected**

Decomposition in strongly connected components

- A subgraph $H \subset G$ is a **strongly connected component** if H is strongly connected and any other subgraph containing H is not
- **Condensation digraph** of G
  1. the nodes are the strongly connected components of G
  2. there exists a directed edge from node H_1 to node H_2 iff there exists a directed edge in G from a node of H_1 to a node of H_2

Properties of the condensation digraph

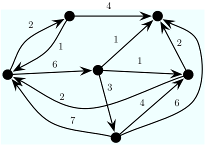
1. every condensation digraph is acyclic
2. G contains a globally reachable node iff $C(G)$ contains a globally reachable node
3. G contains a directed spanning tree iff $C(G)$ contains a directed spanning tree

Weighted digraphs

A **weighted digraph** is a triplet $G = (V, E, A)$, where (V, E) is a digraph and A is an $n \times n$ **weighted adjacency matrix** such that

$$a_{ij} > 0 \text{ if } (v_i, v_j) \text{ is an edge of } G, \text{ and } a_{ij} = 0 \text{ otherwise}$$

Scalars a_{ij} are weights for the edges of G. Weighted digraph is undirected if $a_{ij} = a_{ji}$ for all $i, j \in \{1, \dots, n\}$



Weighted digraphs: cont'd

Weighted **out-degree** and **in-degree**

$$d_{\text{out}}(i) = \sum_{j=1}^{n} a_{ij} \quad \text{and} \quad d_{\text{in}}(i) = \sum_{j=1}^{n} a_{ji}$$

G is **weight-balanced** if each vertex has equal in- and out-degree

Weighted out-degree diagonal matrix $D_{\text{out}}(G)$: $(D_{\text{out}}(G))_{ii} = d_{\text{out}}(i)$

Weighted in-degree diagonal matrix $D_{\text{in}}(G)$: $(D_{\text{in}}(G))_{ii} = d_{\text{in}}(i)$

Algebraic Graph Theory

- motivating example: linear averaging
- when is certain matrix primitive
- so far, graph theory: connectivity and periodicity
- next, how to relate graphs to matrices

Properties of the adjacency matrix

- G is weighted digraph of order n
- A is weighted adjacency matrix
- D_{out} is weighted out-degree matrix

Weight-balanced digraph $\rightsquigarrow$ doubly stochastic adjacency matrix

$$F = \begin{cases} D_{\text{out}}^{-1} A, & \text{if each out-degree is positive,} \\ (I_n + D_{\text{out}})^{-1}(I_n + A), & \text{otherwise.} \end{cases}$$

1. F is row-stochastic; and
2. F is doubly stochastic if G is weight-balanced and the weighted degree is constant for all vertices.

Properties of the adjacency matrix: cont'd

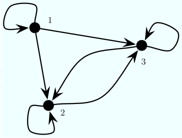
- $A_{0,1} \in \{0,1\}^{n\times n}$ is unweighted adjacency matrix
- G possibly contains self-loops

Directed paths in digraph $\rightsquigarrow$ powers of the adjacency matrix

For all $i, j, k \in \{1, \dots, n\}$

1. the (i,j) entry of $A_{0,1}^k$ equals the number of directed paths of length k (including paths with self-loops) from node i to node j
2. the (i,j) entry of A^k is positive if and only if there exists a directed path of length k (including paths with self-loops) from node i to node j.

Properties of the adjacency matrix: cont'd



$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- vertices 2 and 3 are globally reachable
- digraph is not strongly connected cause vertex 1 has no in-neighbor other than itself
- adjacency matrix is reducible

Properties of the adjacency matrix: cont'd

Digraph connectivity ⇝ powers of adjacency matrix

The following statements are equivalent:

1. G is strongly connected,
2. A is irreducible; and
3. $\sum_{k=0}^{n-1} A^k$ is positive.

For any $j \in \{1, \dots, n\}$, the following statements are equivalent:

4. the jth node of G is globally reachable; and
5. the jth column of $\sum_{k=0}^{n-1} A^k$ has positive entries.

Properties of the adjacency matrix: cont'd

Digraph connectivity ⇝ powers of adjacency matrix: cont'd

Assume self-loops at each node.

The following statements are equivalent:

4. G is strongly connected; and
5. A^{n-1} has positive entries.

For any $j \in \{1, \dots, n\}$, the following two statements are equivalent:

4. the jth node of G is globally reachable; and
5. the jth column of A^{n-1} has positive entries.

Properties of the adjacency matrix: final

- G is weighted digraph of order n
- A is weighted adjacency matrix

Strongly connected + aperiodic digraph = primitive adjacency matrix

The following two statements are equivalent:

1. G is strongly connected and aperiodic; and
2. A is primitive, i.e., there exists $k \in \mathbb{N}$ such that A^k is positive.

Algebraic Graph Theory: the Laplacian matrix

The **graph Laplacian** of the weighted digraph G is

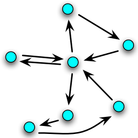
$$L(G) = D_{\text{out}}(G) - A(G)$$

Properties of the Laplacian matrix

The following statements hold:

1. $L(G)\mathbf{1}_n = \mathbf{0}$
2. G is undirected iff $L(G)$ is symmetric
3. if G is undirected, then $L(G)$ is positive semidefinite
4. G contains a globally reachable vertex iff $\operatorname{rank} L(G) = n - 1$
5. G is weight-balanced iff $\mathbf{1}_n^T L(G) = \mathbf{0}$

Disagreement function



Disagreement function

$$\Phi_G(x) = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}(x_j - x_i)^2$$

If G weight-balanced,

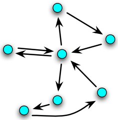
- $\Phi_G(x) = x^T L(G) x$

If G weight-balanced and weakly connected,

- $\lambda_n(\mathsf{Sym}(L))\,\|x - \text{Ave}(x)\mathbf{1}_n\|^2 \geq \Phi_G(x) \geq \lambda_2(\mathsf{Sym}(L))\,\|x - \text{Ave}(x)\mathbf{1}_n\|^2$

Linear distributed iterations

Data exchange and fusion is a basic task for any network



Given graph $G = (\{1,\dots,n\}, E_{\text{cmm}})$, matrix $F = (f_{ij}) \in \mathbb{R}^{n\times n}$ is **compatible** if

$$f_{ij} \neq 0 \quad \text{if and only if} \quad (j,i) \in E_{\text{cmm}}$$

Given compatible F, LINEAR COMBINATION algorithm, starting from $w(0) \in \mathbb{R}^n$, is

$$w(\ell+1) = F \cdot w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0}$$

In coordinates,

$$w_i(\ell+1) = f_{ii}w_i(\ell) + \sum_{j\in\mathcal{N}^{\text{in}}(i)} f_{ij}w_j(\ell)$$

Time-dependent linear iterations

Discrete-time linear dynamical systems represent an important class of iterative algorithms with applications in

- optimization
- systems of equations
- distributed decision making

Linear combination procedure can be extended to sequence of time-dependent state-transition functions associated with $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n\times n}$,

$$w(\ell+1) = F(\ell)\cdot w(\ell), \quad \ell \in \mathbb{Z}_{\geq 0} \quad \text{and} \quad w(0) \in \mathbb{R}^n$$

Linear averaging over switching graphs: flocking example

Consider a group of agents in the plane moving with unit speed and adjusting their heading as follows:

> *at integer instants of time, each agent senses the heading of its neighbors (other agents within some specified distance r), and re-sets its heading to the average of its own heading and its neighbors' heading*

Mathematically, if (x_i, y_i) is position of agent i,

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \quad |v_i| = 1 \tag{1}$$

$$\begin{aligned}\theta_i(\ell+1) &= \frac{1}{1+|\mathcal{N}_i|}\Big(\theta_i(\ell) + \sum_{j\in\mathcal{N}_i}\theta_j(\ell)\Big) \\ &= \mathsf{average}(\theta_i(\ell), \theta_j(\ell) \text{ for all in-neighbors } j\})\end{aligned}$$

Topology might change from one time instant to the next

Averaging algorithms

A **(distributed) averaging algorithm** is a linear algorithm associated to a **(row) stochastic** matrix $F \in \mathbb{R}^{n\times n}$

$$\sum_{j=1}^{n} f_{ij} = 1 \quad \text{and} \quad f_{ij} \geq 0 \quad \text{for all } i, j \in \{1,\dots,n\}$$

Note: $F \cdot \mathbf{1}_n = \mathbf{1}_n$. The vector subspace generated by $\mathbf{1}_n$ is the diagonal set $\mathsf{diag}(\mathbb{R}^n)$ of $\mathbb{R}^n$. Points in $\mathsf{diag}(\mathbb{R}^n)$ are **agreeement configurations**

An algorithm **achieves agreement** if it steers the network state towards the set of agreement configurations

Laplacian- or adjacency-based agreement

Let $G = (\{1,\dots,n\}, E_{\text{cmm}}, A)$ be weighted digraph

Laplacian-based:

$$w(\ell+1) = (I_n - \varepsilon L(G)) \cdot w(\ell)$$

where $0 < \varepsilon \leq \min_i\{1/d_{\text{out}}(i)\}$ to have $I_n - \varepsilon L(G)$ stochastic

Adjacency-based:

$$w(\ell+1) = (I_n + D_{\text{out}}(G))^{-1}(I_n + A(G)) \cdot w(\ell)$$

resulting stochastic matrix has always non-zero diagonal entries

- Any averaging algorithm may be written as Laplacian- or adjacency-based
- If G is unweighted, undirected, and without self-loops, then **adjacency-based averaging** = **equal-neighbor rule** = **Vicsek's model**

$$w_i(\ell+1) = \mathsf{average}\Big(w_i(\ell), \{w_j(\ell) \mid j \in \mathcal{N}_G(i)\}\Big)$$

Stability of agreement configurations

Consider a sequence of stochastic matrices $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n\times n}$:

- $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ is **non-degenerate** if there exists $\alpha \in \mathbb{R}_{>0}$ such that, for all $\ell \in \mathbb{Z}_{\geq 0}$,

$$f_{ii}(\ell) \geq \alpha, \text{ for all } i \in \{1,\dots,n\} \quad \text{and} \quad f_{ij}(\ell) \in \{0\} \cup [\alpha, 1], \text{ for all } i \neq j \in \{1,\dots,n\}$$

- for $\ell \in \mathbb{Z}_{\geq 0}$, let $G(\ell)$ be the unweighted graph associated to $F(\ell)$

Stability – directed case

Theorem

Let $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\}$ be a non-degenerate sequence of stochastic matrices. The following are equivalent:

1. *the set $\mathsf{diag}(\mathbb{R}^n)$ is globally attractive for the averaging algorithm*
2. *there exists a duration $\delta \in \mathbb{N}$ such that, for all $\ell \in \mathbb{Z}_{\geq 0}$, the digraph*

$$G(\ell + 1) \cup \cdots \cup G(\ell + \delta)$$

contains a globally reachable vertex.

In other words, the linear algorithm converges uniformly and asymptotically to the vector subspace generated by $\mathbf{1}_n$

Stability – undirected case

Theorem

Let $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n\times n}$ be a non-degenerate sequence of stochastic, symmetric matrices. The following are equivalent:

1. *the set $\mathsf{diag}(\mathbb{R}^n)$ is globally attractive for the averaging algorithm*
2. *for all $\ell \in \mathbb{Z}_{\geq 0}$, the following graph is connected*

$$\bigcup_{\tau \geq \ell} G(\tau)$$

In both results, each individual evolution converges to an specific point of $\mathsf{diag}(\mathbb{R}^n)$, rather than converging to the whole set

Non-degeneracy requirement in both results can not be removed to achieve agreement

Laplacian- and adjancency-based agreement
Convergence

The following statements are equivalent

1. Laplacian-based agreement algorithm is globally attractive with respect to $\mathsf{diag}(\mathbb{R}^n)$
2. Adjancency-based agreement algorithm is globally attractive with respect to $\mathsf{diag}(\mathbb{R}^n)$
3. G contains a globally reachable node

Time-independent averaging algorithm

Consider the time-invariant linear system on $\mathbb{R}^n$

$$w(\ell + 1) = Fw(\ell) \tag{2}$$

Theorem (Time-independent averaging algorithm)

Assume

- *$F \in \mathbb{R}^{n\times n}$ is stochastic*

  $G(F)$ denotes associated weighted digraph
- *$v \in \mathbb{R}^n$ is a left eigenvector of F with eigenvalue 1*
- *assume either one of the two following properties:*
  1. *F is primitive (i.e., $G(F)$ is strongly connected and aperiodic); or*
  2. *F has non-zero diagonal terms and a column of F^{n-1} has positive entries (i.e., $G(F)$ has self-loops at each node and has a globally reachable node).*

Then every trajectory converges to $(v^T w(0)/v^T \mathbf{1}_n)\mathbf{1}_n$.

What is the agreement value?

Specific value upon which all w_i, $i \in \{1,\dots,n\}$ agree is **unknown** – complex function of initial condition and specific sequence of matrices

Given time-dependent **doubly stochastic** $\{F(\ell) \mid \ell \in \mathbb{Z}_{\geq 0}\} \subset \mathbb{R}^{n\times n}$ satisfying assumptions for convergence (direct or undirect, time-invariant), then

$$\sum_{i=1}^{n} w_i(\ell+1) = \mathbf{1}_n^T w(\ell+1) = \mathbf{1}_n^T F(\ell) w(\ell) = \mathbf{1}_n^T w(\ell) = \sum_{i=1}^{n} w_i(\ell)$$

Since in the limit all entries of w must coincide, **average-consensus**

$$\lim_{\ell\to+\infty} w_j(\ell) = \frac{1}{n}\sum_{i=1}^{n} w_i(0), \quad j \in \{1,\dots,n\}$$

Synchronous networks

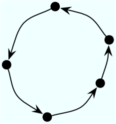
Previous examples of linear distributed iterations are particular class of algorithms that can be run in parallel by network of computers

Theory of parallel computing and distributed algorithms studies general classes of algorithms that can be implemented in static networks (neighboring relationships do not change)

Synchronous network: cont'd

Synchronous network is group of processors with ability to exchange messages and perform local computations. Mathematically, a digraph (I, E_{cmm}),

1. $I = \{1,\dots,n\}$ is the **set of unique identifiers (UIDs)**, and
2. E_{cmm} is a set of directed edges over the vertices I, called the communication links



Distributed algorithm

Distributed algorithm $\mathcal{DA}$ for a network $\mathcal{S}$ consists of the sets

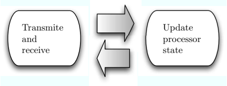
1. $\mathbb{A}$, a set containing the `null` element, called the **communication alphabet**; elements of $\mathbb{A}$ are called **messages**;
2. $W^{[i]}$, $i \in I$, called the **processor state sets**;
3. $W_0^{[i]} \subseteq W^{[i]}$, $i \in I$, sets of **allowable initial values**;

and of the maps

1. $\mathrm{msg}^{[i]} : W^{[i]} \times I \to \mathbb{A}$, $i \in I$, called **message-generation functions**;
2. $\mathrm{stf}^{[i]} : W^{[i]} \times \mathbb{A}^n \to W^{[i]}$, $i \in I$, called **state-transition functions**.

If $W^{[i]} = W$, $\mathrm{msg}^{[i]} = \mathrm{msg}$, and $\mathrm{stf}^{[i]} = \mathrm{stf}$ for all $i \in I$, then $\mathcal{DA}$ is said to be **uniform** and is described by a tuple $(\mathbb{A}, W, \{W_0^{[i]}\}_{i\in I}, \mathrm{msg}, \mathrm{stf})$

Network evolution



Discrete-time communication and computation: evolution of $(\mathcal{S}, \mathcal{DA})$ from initial conditions $w_0^{[i]} \in W_0^{[i]}$ is the collection of trajectories $w^{[i]} : \mathbb{Z}_{\geq 0} \to W^{[i]}$ satisfying

$$w^{[i]}(\ell) = \text{stf}^{[i]}(w^{[i]}(\ell - 1), y^{[i]}(\ell))$$

where $w^{[i]}(-1) = w_0^{[i]}$, $i \in I$, and $y^{[i]} : \mathbb{Z}_{\geq 0} \to \mathbb{A}^n$ are the messages received by processor i:

$$y_j^{[i]}(\ell) = \begin{cases} \text{msg}^{[j]}(w^{[j]}(\ell-1), i), & \text{if } (i,j) \in E_{\text{cmm}}, \\ \texttt{null}, & \text{otherwise.} \end{cases}$$

Complexity notions

How good is a distributed algorithm? How costly to execute?
Complexity notions characterize performance of distributed algorithms

Algorithm completion: an algorithm **terminates** when only `null` messages are transmitted and all processors states become constants

Time complexity: $\mathsf{TC}(\mathcal{DA}, \mathcal{S})$ is maximum number of rounds required by execution of $\mathcal{DA}$ on $\mathcal{S}$ among all allowable initial states

Space complexity: $\mathsf{SC}(\mathcal{DA}, \mathcal{S})$ is maximum number of basic memory units required by a processor executing $\mathcal{DA}$ on $\mathcal{S}$ among all processors and all allowable initial states

Communication complexity: $\mathsf{CC}(\mathcal{DA}, \mathcal{S})$ is maximum number of basic messages transmitted over the entire network during execution of $\mathcal{DA}$ among all allowable initial states

until termination (basic memory unit, message contains $\log(n)$ bits)

Leader election by comparison

Problem

Assume that all processors of a network have a state variable, say `leader`*, initially set to* `unknown`

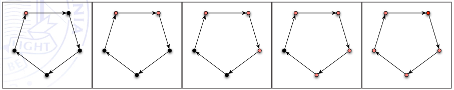
A leader is elected when one and only one processor has the state variable set to `true` *and all others have it set to* `false`

Elect a leader

Le Lann-Chang-Roberts (LCR) algorithm solves leader election in rings with **complexities**

1. time complexity n
2. space complexity 2
3. communication complexity $\Theta(n^2)$

The LCR algorithm: informal description



1. First frame: the agent with the maximum UID is colored in red.
2. After 5 communication rounds, this agent receives its own UID from its in-neighbor and declares itself the leader.

The LCR algorithm

Network: Ring network
Alphabet: $\mathbb{A} = \{1, \dots, n\} \cup \{\texttt{null}\}$
Processor State: $w = (\texttt{my-id}, \texttt{max-id}, \texttt{leader}, \texttt{snd-flag})$, where

my-id	$\in \{1, \dots, n\}$,	initially:	$\texttt{my-id}^{[i]} = i$ for all i
max-id	$\in \{1, \dots, n\}$,	initially:	$\texttt{max-id}^{[i]} = i$ for all i
leader	$\in \{\texttt{true}, \texttt{unknown}\}$,	initially:	$\texttt{leader}^{[i]} = \texttt{unknown}$ for all i
snd-flag	$\in \{\texttt{true}, \texttt{false}\}$,	initially:	$\texttt{snd-flag}^{[i]} = \texttt{true}$ for all i

```
function msg(w, i)
if snd-flag = true then
    return max-id
else
    return null
```


The LCR algorithm

```
function stf(w, y)
case
  (y contains only null msgs) OR (largest identifier in y < my-id):
      new-id := max-id
      new-lead := leader
      new-snd-flag := false
  (largest identifier in y = my-id):
      new-id := max-id
      new-lead := true
      new-snd-flag := false
  (largest identifier in y > my-id):
      new-id := largest identifier in y
      new-lead := false
      new-snd-flag := true
return (my-id, new-id, new-lead, new-snd-flag)
```


Quantifying time, space, and communication complexity

Asymptotic "order of magnitude" measures. E.g., algorithm has **time complexity of order**

1. $\Omega(f(n))$ if, for all n, $\exists$ network of order n and initial processor values such that TC is greater than a constant factor times $f(n)$
2. $O(f(n))$ if, for all n, for all networks of order n and for all initial processor values, TC is lower than a constant factor times $f(n)$
3. $\Theta(f(n))$ if TC is of order $\Omega(f(n))$ and $O(f(n))$ at the same time

Similar conventions for space and communication complexity

Numerous variations of complexity definitions are possible

1. "Global" rather than "existential" lower bounds
2. Expected or average complexity notions
3. Complexity notions for problems, rather than for algorithms

Summary and conclusions

A primer on graph theory

1. Basic graph-theoretic notions and connectivity notions
2. Adjacency and Laplacian matrices

Linear distributed iterations

1. Discrete-time linear dynamical systems
2. averaging algorithms and convergence results

Introduction to distributed algorithms

1. Model
2. Complexity notions
3. Leader election

References

Graph theory

- R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2000
- C. D. Godsil and G. F. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer, 2001
- N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 2 edition, 1994
- R. Merris. Laplacian matrices of a graph: A survey. *Linear Algebra its Applications*, 197:143--176, 1994
- E. Seneta. *Non-negative Matrices and Markov Chains*. Springer, 2 edition, 1981

Distributed algorithms

- N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997
- D. Peleg. *Distributed Computing. A Locality-Sensitive Approach*. Monographs on Discrete Mathematics and Applications. SIAM, 2000

References: cont'd

Linear distributed iterations and agreement algorithms

- M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118--121, 1974
- H. J. Landau and A. M. Odlyzko. Bounds for eigenvalues of certain stochastic matrices. *Linear Algebra and its Applications*, 38:5--15, 1981
- R. Cogburn. The ergodic theory of Markov chains in random environments. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 66(1):109--128, 1984
- G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279--301, 1989
- J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Massachusetts Institute of Technology, November 1984. Technical Report LIDS-TH-1424. Available electronically at http://web.mit.edu/jnt/www/Papers/PhD-84-jnt.pdf

References: cont'd

Linear distributed iterations and agreement algorithms: cont'd

- D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997
- A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988--1001, 2003
- R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520--1533, 2004
- V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *IEEE Conf. on Decision and Control and European Control Conference*, pages 2996--3000, Seville, Spain, December 2005
- L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169--182, 2005