Distributed average consensus for wireless networks

Philippe Ciblat

Joint work with P. Bianchi, W. Hachem and F. Iutzeler

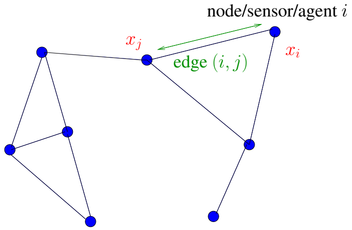
Institut Mines-Telecom/Telecom ParisTech, Paris, France

Outline

1. A few applications
2. Standard (single-variate based-) algorithms
   - Synchronous case [DeGroot1974,Tsitsiklis1984]
   - Asynchronous case
     - doubly-stochastic case [Boyd2006]
     - row-stochastic case [Aysal2009]
     - column-stochastic case : can take benefit of broadcast nature of wireless channel
3. New (bivariate based-) algorithm [Iutzeler2013]
4. Numerical illustrations
5. Perspectives : distributed optimization

Part 1 : Applications

Example 1 : distributed monitoring in sensor networks



- N nodes/sensors/agents
- x_i measurement at node i
- **Applications :** practical measurements of temperature, gas pressure, ...

Goal

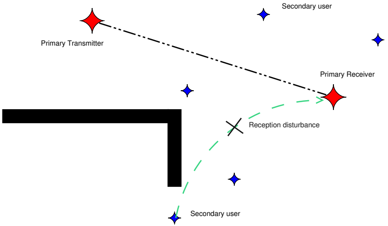
At each node, we want to compute $x_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N} x_i$

but

- No fusion center
- Only local communications allowed

distributed average computation is needed

Example 2 : fully-distributed spectrum sensing for cognitive radio



Problem : a secondary user is disturbing the primary receiver (hidden terminal problem)

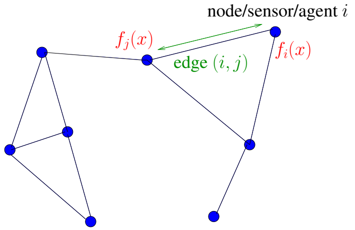
Solution : secondary users have to cooperate to detect the primary user

Goal : Optimal test for Gaussian signals

$$T = \frac{1}{N}\sum_{i=1}^{N} x_i \gtrless \eta, \quad \text{with} \quad x_i \propto \sum_{k=1}^{N_s} |y_i(k)|^2$$

distributed average computation is needed

Example 3 : distributed quadratic form optimization



$$x_i = x + e_i$$

with

- x_i measurement at node i
- x common **unknown** parameter
- e_i white Gaussian noise
- **Applications :** localization, ...

Goal : Maximum-Likelihood estimator

The best estimator for x is

$$\hat{x}_{\text{opt}} = \arg\min_x f(x) \triangleq \sum_{i=1}^{N} f_i(x) \quad \text{with} \quad f_i(x) = (x_i - x)^2$$

$$\hat{x}_{\text{opt}} = x_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{N} x_i \Leftrightarrow$$ **distributed average computation is needed**

Example 4 : rendez-vous problem

- Let $x_i(0) = x_i$ be the initial position of node i
- At time $(k+1)$, for each node i

$$x_i(k+1) = x_i(k) + \alpha \sum_{j \in \mathcal{N}_i} (x_j(k) - x_i(k))$$

with

- $x_i(k)$ position of node i at time k
- $\mathcal{N}_i$ neighborhood of node i
- α a positive parameter

- **Applications :** flocking analysis, ...

Goal

- Is there consensus ($\lim_{k\to\infty} x_i(k) = c$) ?
- Which condition on α for satisfying $c = x_{\mathrm{ave}}$?

Part 2 : Standard (single-variate based-) algorithms

Algorithmic model

Principle

- Updates correspond to a positive linear combination of previous values
- Which nodes wake up at a given time ?
- Which nodes take part to linear combination ?

Synchronous case :

- each node wakes up at each time and performs the same combination
- **No randomness**

Asynchronous case :

- a set of nodes wakes up at a given time randomly
- these selected nodes perform random linear combinations
- model very general (e.g. deterministic algos with collision or random environment, ...)
- **Randomness**

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on K

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on **K**

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on K

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on $\mathbf{K}$

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on **K**

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on K

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

$\Rightarrow$ **Row-stochastic matrix**

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on **K**

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

$\Rightarrow$ **Column-stochastic matrix**

Sync. case : Rendez-vous algorithm [Tsitsiklis1984]

$$\mathbf{x}(k+1) = \mathbf{K}\mathbf{x}(k)$$

with

- $\mathbf{x}(k) = [x_1(k), \cdots, x_N(k)]^{\mathrm{T}}$ where $x_i(k)$ value of node i at time k
- $\mathbf{x}(0) = [x_1, \cdots, x_N]^{\mathrm{T}}$ are the initial measurements.
- matrix $\mathbf{K}$ such that

$$\mathbf{K}_{i,j} = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i \\ 1 - \alpha d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

where d_i is the number of neighbors of node i.

Property on K

- If symmetric links (link $i \to j$ exists iff link $j \to i$ exists)
- If $\alpha < 1/\max_i\{d_i\}$

Then

- Non-negative
- Positive diagonal
- Row sum $= 1$
- Column sum $= 1$

$\Rightarrow$ **Doubly-stochastic matrix**

Sync. case : Metropolis-Hastings algorithm [DeGroot1974]

First algorithm developed for distributed average consensus

$$\mathbf{K}_{i,j} = \begin{cases} \frac{1}{1+\max\{d_i,d_j\}} & \text{if } j \in \mathcal{N}_i \\ 1 - \sum_{j\in\mathcal{N}_i} \mathbf{K}_{i,j} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

- Non-negative : positive combination
- Positive diagonal : each node keeps its own information
- Row sum $= 1$: barycenter of received current values

Consensus conservation $\Leftrightarrow$ $\mathbf{x}(k) = c\mathbf{1}$, so $\mathbf{x}(k+1) = c\mathbf{1}$

- Column sum $= 1$: ? ? ?

Average conservation $\Leftrightarrow$ $\mathbf{1}^{\mathrm{T}}\mathbf{x}(k+1) = \mathbf{1}^{\mathrm{T}}\mathbf{x}(k)$

Convergence result

$$\mathbf{x}(k) = \mathbf{K}^k \mathbf{x}(0) \quad \text{with } \mathbf{K} \text{ doubly-stochastic}$$

One tool : primitive matrix

- **M** primitive matrix iff non-negative **and** $\exists\, m \geq 1$ s. t. $\mathbf{M}^m$ positive matrix
- This property holds if nodes' communication graph is connected

Any $N \times N$ row-stochastic matrix is a transition probability matrix of a discrete-time Markov Chain with N states and conversely.

If **K** primitive and column-stochastic, then

$$\lim_{k\to\infty} \mathbf{K}^k = \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathrm{T}} \triangleq \mathbf{J} \quad (\text{projector on span}(\mathbf{1}))$$

and

$$\mathbf{x}(k) \to x_{\text{ave}}\mathbf{1} \text{ when } k \to \infty$$

Convergence speed result

$$\mathbf{x}(k) = \mathbf{J}\mathbf{x}(k) + \mathbf{J}^{\perp}\mathbf{x}(k) \overset{\mathbf{K}\ col.-sto.}{=} x_{\text{ave}}\mathbf{1} + \mathbf{J}^{\perp}\mathbf{x}(k)$$

Convergence of $\mathbf{x}(k) \sim$ convergence of $\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2$

In addition,

$$\|\mathbf{J}^{\perp}\mathbf{x}(k+1)\|^2 \overset{\mathbf{K}\ row.-sto.}{\leq} \rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right)\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2.$$

Is $\rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right) < 1$?

Convergence speed result

$$\mathbf{x}(k) = \mathbf{J}\mathbf{x}(k) + \mathbf{J}^{\perp}\mathbf{x}(k) \overset{\mathbf{K}\ col.-sto.}{=} x_{\text{ave}}\mathbf{1} + \mathbf{J}^{\perp}\mathbf{x}(k)$$

Convergence of $\mathbf{x}(k) \sim$ convergence of $\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2$

In addition,

$$\|\mathbf{J}^{\perp}\mathbf{x}(k+1)\|^2 \overset{\mathbf{K}\ row.-sto.}{\leq} \rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right)\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2.$$

Is $\rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right) < 1$? YES (since K primitive and $\mathbf{K}\mathbf{K}^{\mathrm{T}}$ as well)

Convergence speed result

$$\mathbf{x}(k) = \mathbf{J}\mathbf{x}(k) + \mathbf{J}^{\perp}\mathbf{x}(k) \overset{\mathbf{K}\ col.-sto.}{=} x_{\text{ave}}\mathbf{1} + \mathbf{J}^{\perp}\mathbf{x}(k)$$

Convergence of $\mathbf{x}(k)$ $\sim$ convergence of $\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2$

In addition,

$$\|\mathbf{J}^{\perp}\mathbf{x}(k+1)\|^2 \overset{\mathbf{K}\ row.-sto.}{\leq} \rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right)\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2.$$

Is $\rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right) < 1$? YES (since K primitive and $\mathbf{K}\mathbf{K}^{\mathrm{T}}$ as well)

Perron-Frobenius theorem

Let $\mathbf{M}$ be a primitive matrix and $\rho(\mathbf{M}) = \max_i |\lambda_i(\mathbf{M})|$ its *spectral radius*. Then

i) $\rho(\mathbf{M})$ is the unique maximum eigenvalue and is simple.

If $\mathbf{M}$ is either row-stochastic or column-stochastic matrix, then

ii) $\rho(\mathbf{M}) = 1$.

iii) $\mathbf{1}/\sqrt{N}$ (right or left)-eigenvector for the eigenvalue 1.

Convergence speed result

$$\mathbf{x}(k) = \mathbf{J}\mathbf{x}(k) + \mathbf{J}^{\perp}\mathbf{x}(k) \overset{\mathbf{K}\ col.-sto.}{=} x_{\text{ave}}\mathbf{1} + \mathbf{J}^{\perp}\mathbf{x}(k)$$

Convergence of $\mathbf{x}(k) \sim$ convergence of $\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2$

In addition,

$$\|\mathbf{J}^{\perp}\mathbf{x}(k+1)\|^2 \overset{\mathbf{K}\ row.-sto.}{\leq} \rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right)\|\mathbf{J}^{\perp}\mathbf{x}(k)\|^2.$$

Is $\rho\left(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}\right) < 1$? YES (since K primitive and $\mathbf{K}\mathbf{K}^{\mathrm{T}}$ as well)

Perron-Frobenius theorem

Let **M** be a primitive matrix and $\rho(\mathbf{M}) = \max_i |\lambda_i(\mathbf{M})|$ its *spectral radius*. Then

i) $\rho(\mathbf{M})$ is the unique maximum eigenvalue and is simple.

If **M** is either row-stochastic or column-stochastic matrix, then

ii) $\rho(\mathbf{M}) = 1$.

iii) $\mathbf{1}/\sqrt{N}$ (right or left)-eigenvector for the eigenvalue 1.

Main result

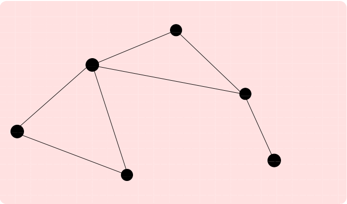
$$\text{MSE}(k) \leq \alpha^k \|\mathbf{J}^{\perp}\mathbf{x}(0)\|^2 \text{ with } \alpha = \rho(\mathbf{K}\mathbf{K}^{\mathrm{T}} - \mathbf{J}) < 1$$

Asynchronous case

$$\mathbf{x}(k+1) = \mathbf{K}_{\xi_{k+1}} \mathbf{x}(k)$$

- $\{\mathbf{K}_{\xi_k}\}_k$ i.i.d. process valued in $\mathcal{K} = \{\mathbf{K}_j\}_{j=1,\ldots,M}$
- $\mathbf{K}_{\xi_k}$ non-negative, positive diagonal

Example : Random Pairwise Gossip [Boyd2006]



At time k, let i be the active node

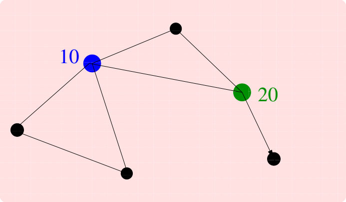
- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Asynchronous case

$$\mathbf{x}(k+1) = \mathbf{K}_{\xi_{k+1}} \mathbf{x}(k)$$

- $\{\mathbf{K}_{\xi_k}\}_k$ i.i.d. process valued in $\mathcal{K} = \{\mathbf{K}_j\}_{j=1,\ldots,M}$
- $\mathbf{K}_{\xi_k}$ non-negative, positive diagonal

Example : Random Pairwise Gossip [Boyd2006]



At time k, let i be the active node

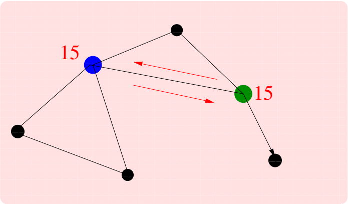
- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Asynchronous case

$$\mathbf{x}(k+1) = \mathbf{K}_{\xi_{k+1}}\mathbf{x}(k)$$

- $\{\mathbf{K}_{\xi_k}\}_k$ i.i.d. process valued in $\mathcal{K} = \{\mathbf{K}_j\}_{j=1,\dots,M}$
- $\mathbf{K}_{\xi_k}$ non-negative, positive diagonal

Example : Random Pairwise Gossip [Boyd2006]



At time k, let i be the active node

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Asynchronous case

$$\mathbf{x}(k+1) = \mathbf{K}_{\xi_{k+1}} \mathbf{x}(k)$$

- $\{\mathbf{K}_{\xi_k}\}_k$ i.i.d. process valued in $\mathcal{K} = \{\mathbf{K}_j\}_{j=1,\ldots,M}$
- $\mathbf{K}_{\xi_k}$ non-negative, positive diagonal

Example : Random Pairwise Gossip [Boyd2006]

$$\mathbf{K}_{\{i,j\}} = \begin{bmatrix} \ddots 1 \ddots & & & \\ & 1/2 & & 1/2 & \\ & & \ddots 1 \ddots & & \\ & 1/2 & & 1/2 & \\ & & & & \ddots 1 \ddots \end{bmatrix}$$

At time k, let i be the active node

- i chooses a neighbor j uniformly in $\mathcal{N}_i$
- i and j exchange their values
- i and j update : $x_i(k+1) = x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Doubly-stochastic update matrices

Example : Random Pairwise Gossip [Boyd2006]

Main results

As doubly-stochastic matrices, simple adaptation of synchronous case

$$\mathbf{x}(k) \overset{a.s.}{\to} x_{\mathrm{ave}}\mathbf{1} \text{ when } k \to \infty$$

$$\mathrm{MSE}(k) \leq \beta^k \|\mathbf{J}^{\perp}\mathbf{x}(0)\|^2 \text{ with } \beta = \rho(\mathbb{E}[\mathbf{K}\mathbf{K}^{\mathrm{T}}] - \mathbf{J}) < 1$$

Two main drawbacks :

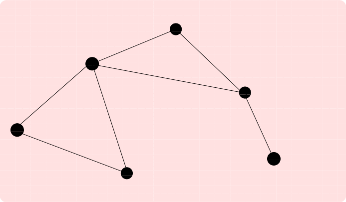
- feedback needed
  - lack of robustness
  - requires slow-varying network (especially if routing)
  - prevents to take benefit of broadcast nature of wireless channel
- broadcast nature of wireless channel not used

Main open question : how broadcasting information without feedback ?

Answer : Iutzeler-Ciblat-Hachem in ICASSP'2012

Example : Broadcast Gossip [Aysal2009]

First algorithm taken into account broadcast nature of wireless channel

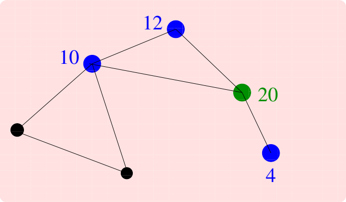


At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Example : Broadcast Gossip [Aysal2009]

First algorithm taken into account broadcast nature of wireless channel

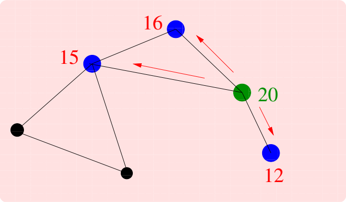


At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Example : Broadcast Gossip [Aysal2009]

First algorithm taken into account broadcast nature of wireless channel



At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Example : Broadcast Gossip [Aysal2009]

First algorithm taken into account broadcast nature of wireless channel

$$\mathbf{K}_i = \begin{bmatrix} 1/2 & & 1/2 & & \\ & \ddots 1 \ddots & & & \\ & & 1 & & \\ & & 1/2 & 1/2 & \\ & & & & \ddots 1 \ddots \end{bmatrix}$$

At time k, let i be the active node

- i broadcasts its value to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $x_j(k+1) = \frac{x_i(k)+x_j(k)}{2}$.

Row-stochastic update matrices

Exemple : Broadcast Gossip [Aysal2009]

- **Problem :** no column-stochastic matrices

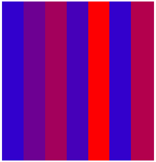
  $\Rightarrow$ no average conservation ; information is lost (Slide 9) ! ! !

- **Result :** $\exists$ $\mathbf{v}$ non-negative vector ($\mathbf{1}^{\mathrm{T}}\mathbf{v} = 1$),

$$\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_2}\mathbf{K}_{\xi_1} \stackrel{k\to\infty}{\longrightarrow} \mathbf{1}\mathbf{v}^{\mathrm{T}}$$

and

$$\mathbf{x}(k) \stackrel{a.s.}{\to} (\mathbf{v}^{\mathrm{T}}\mathbf{x}(0))\mathbf{1}$$



  - Nodes reach consensus, but not the good one
  - If $\mathbf{K}_i$ column-stochastic as well, $\mathbf{v} = (1/N)\mathbf{1}$, the average is reached

Conclusion

Avoid to do that ! ! !

How using broadcast channel and converging as well ?

Doubly-stochastic update matrices are good candidates ... (Slide 13)

Let us try to build update matrices with broadcast and without feedback

- node 4 broadcasts information to its neighbors 2 and 3.
- **no feedback**.
- Row-stochasticity is possible ... but not double-stochasticity
- Column-stochasticity is possible ($k_2 + k_3 + k_4 = 1$) ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & k_{2,2} & 0 & k_{2,4} \\ 0 & 0 & k_{3,3} & k_{3,4} \\ 0 & 0 & 0 & k_{4,4} \end{bmatrix}$$

How using broadcast channel and converging as well ?

Doubly-stochastic update matrices are good candidates ... (Slide 13)

Let us try to build update matrices with broadcast and without feedback

- node 4 broadcasts information to its neighbors 2 and 3.
- **no feedback**.
- Row-stochasticity is possible ... but not double-stochasticity
- Column-stochasticity is possible ($k_2 + k_3 + k_4 = 1$) ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & k_{2,2} & 0 & k_{2,4} \\ 0 & 0 & k_{3,3} & k_{3,4} \\ 0 & \mathbf{0} & \mathbf{0} & k_{4,4} \end{bmatrix}$$

How using broadcast channel and converging as well ?

Doubly-stochastic update matrices are good candidates ... (Slide 13)

Let us try to build update matrices with broadcast and without feedback

- node 4 broadcasts information to its neighbors 2 and 3.
- **no feedback**.
- Row-stochasticity is possible ... but not double-stochasticity
- Column-stochasticity is possible ($k_2 + k_3 + k_4 = 1$) ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 1-k_2 \\ 0 & 0 & k_3 & 1-k_3 \\ 0 & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix}$$

How using broadcast channel and converging as well ?

Doubly-stochastic update matrices are good candidates ... (Slide 13)

Let us try to build update matrices with broadcast and without feedback

- node 4 broadcasts information to its neighbors 2 and 3.
- **no feedback**.
- Row-stochasticity is possible ... but not double-stochasticity
- Column-stochasticity is possible ($k_2 + k_3 + k_4 = 1$) ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & k_2 \\ 0 & 0 & 1 & k_3 \\ 0 & \mathbf{0} & \mathbf{0} & k_4 \end{bmatrix}$$

How using broadcast channel and converging as well ?

Doubly-stochastic update matrices are good candidates ... (Slide 13)

Let us try to build update matrices with broadcast and without feedback

- node 4 broadcasts information to its neighbors 2 and 3.
- **no feedback**.
- Row-stochasticity is possible ... but not double-stochasticity
- Column-stochasticity is possible ($k_2 + k_3 + k_4 = 1$) ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & k_2 \\ 0 & 0 & 1 & k_3 \\ 0 & \mathbf{0} & \mathbf{0} & k_4 \end{bmatrix}$$

... but require FEEDBACK

How using broadcast channel and converging as well ?

Doubly-stochastic update matrices are good candidates ... (Slide 13)

Let us try to build update matrices with broadcast and without feedback

- node 4 broadcasts information to its neighbors 2 and 3.
- **no feedback**.
- Row-stochasticity is possible ... but not double-stochasticity
- Column-stochasticity is possible ($k_2 + k_3 + k_4 = 1$) ... but not double-stochasticity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & k_2 \\ 0 & 0 & 1 & k_3 \\ 0 & \mathbf{0} & \mathbf{0} & k_4 \end{bmatrix}$$

... but require FEEDBACK

Two cases :

- Row-stochastic : **bad** (Slide 15)
- Column-stochastic : **good ?**

Column-stochastic updates matrices

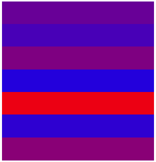
- **It seems worse ! ! !**

- **Result :** $\exists\ \mathbf{v}(k)$ non-negative vectors (with $\mathbf{1}^{\mathrm{T}}\mathbf{v}(k) = 1$) s.t.

$$\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_2}\mathbf{K}_{\xi_1} \sim \mathbf{v}(k)\mathbf{1}^{\mathrm{T}}$$

and

$$\mathbf{x}(k) \sim (N x_{\mathrm{ave}})\mathbf{v}(k)$$



  - **no consensus anymore** since $x_i(k) \propto v_i(k)$
  - **average is available somewhere since** $x_i(k) \propto x_{\mathrm{ave}}$ **but hidden** (Slide 9)

Conclusion

- information is available since average conservation but hidden
- **How recovering it ? How removing** $v_i(k)$ **?** $\Rightarrow$ **side information**

Column-stochastic updates matrices

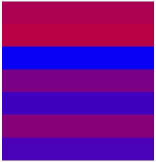
- **It seems worse ! ! !**

- **Result :** $\exists$ $\mathbf{v}(k)$ non-negative vectors (with $\mathbf{1}^{\mathrm{T}}\mathbf{v}(k) = 1$) s.t.

$$\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_2}\mathbf{K}_{\xi_1} \sim \mathbf{v}(k)\mathbf{1}^{\mathrm{T}}$$

and

$$\mathbf{x}(k) \sim (Nx_{\mathrm{ave}})\mathbf{v}(k)$$



  - **no consensus anymore** since $x_i(k) \propto v_i(k)$
  - **average is available somewhere since $x_i(k) \propto x_{\mathrm{ave}}$ but hidden** (Slide 9)

Conclusion

- information is available since average conservation but hidden
- **How recovering it ? How removing $v_i(k)$? $\Rightarrow$ side information**

Column-stochastic updates matrices

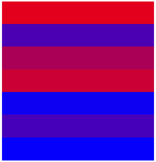
- **It seems worse ! ! !**

- **Result :** $\exists$ $\mathbf{v}(k)$ non-negative vectors (with $\mathbf{1}^{\mathrm{T}}\mathbf{v}(k) = 1$) s.t.

$$\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_2}\mathbf{K}_{\xi_1} \sim \mathbf{v}(k)\mathbf{1}^{\mathrm{T}}$$

and

$$\mathbf{x}(k) \sim (Nx_{\mathrm{ave}})\mathbf{v}(k)$$



  - **no consensus anymore** since $x_i(k) \propto v_i(k)$
  - **average is available somewhere since $x_i(k) \propto x_{\mathrm{ave}}$ but hidden** (Slide 9)

Conclusion

- information is available since average conservation but hidden
- **How recovering it ? How removing $v_i(k)$? $\Rightarrow$ side information**

Part 3 : New (bivariate based-) algorithm

Algorithm principle

An other variable providing $v_i(k)$ has to be computed in parallel

Let $\mathbf{w}(k)$ be this additional variable. We have easily

$$\mathbf{w}(k) \sim N\mathbf{v}(k), \quad \text{if} \quad \mathbf{w}(0) = \mathbf{1} \text{ and } \mathbf{w}(k+1) = \underbrace{\mathbf{K}_{\xi_{k+1}}}_{\text{same as for first variable}} \mathbf{w}(k)$$

Sum-Weight algorithms [Kempe2003, Benezit2010, Iutzeler2013]

Let $\mathbf{s}(0) = \mathbf{x}(0)$ (sum) and $\mathbf{w}(0) = \mathbf{1}$ (weight). At time k, we have

$$\begin{aligned} \mathbf{s}(k+1) &= \mathbf{K}_{\xi_{k+1}} \mathbf{s}(k) \\ \mathbf{w}(k+1) &= \mathbf{K}_{\xi_{k+1}} \mathbf{w}(k) \\ \mathbf{x}(k+1) &= \mathbf{s}(k+1) \oslash \mathbf{w}(k+1) \Leftrightarrow x_i(k+1) = \frac{s_i(k+1)}{w_i(k+1)} \end{aligned}$$

Remarks :

- [Kempe2003] convergence speed for one special algorithm
- [Benezit2010] convergence proof, no convergence speed

Main contributions

Our results [Iutzeler2013]

- Convergence speed for any Sum-Weight algorithm
- Sum-Weight algorithm using broadcast nature of wireless channel

$$\mathbf{x}(k) = x_{\text{ave}}\mathbf{1} + \left((\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1})\mathbf{J}^{\perp}\mathbf{x}(0)\right) \oslash \mathbf{w}(k) \quad (\mathbf{K}_i \text{ col.-sto.})$$

So

$$\|\mathbf{x}(k) - x_{\text{ave}}\mathbf{1}\|_2^2 \leq \Psi_1(k)\Psi_2(k)$$

with

$$\Psi_1(k) = \frac{\|\mathbf{x}(0)\|_2^2}{[\min_i w_i(k)]^2} \quad \text{and } \Psi_2(k) = \left\|(\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1})\mathbf{J}^{\perp}\right\|_{\text{F.}}^2$$

Two main steps :

- $\mathbf{w}(k)$ not often close to 0 through analysis of $\Psi_1(k)$
- $(\mathbf{K}_{\xi_k} \cdots \mathbf{K}_{\xi_1})\mathbf{J}^{\perp}\mathbf{x}(0)$ vanishes for any $\mathbf{x}(0)$ through analysis of $\Psi_2(k)$

Main contributions (cont'd)

Theorem on $\Psi_1(k)$

There is a constant $C < \infty$ and a increasing sequence ($\to \infty$) τ_n s.t.

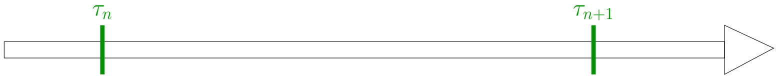
$$\Psi_1(\tau_n) \leq C$$

where

- $\Delta_n = \tau_n - \tau_{n-1}$ is positive i.i.d. geometrically distributed r.v.
- So the event $\{\Psi_1(k) \leq C\}$ occurs infinitely often almost surely.

Theorem on $\Psi_2(k)$

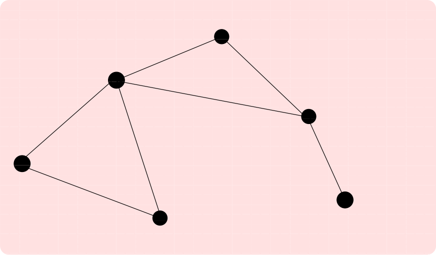
$$\mathbb{E}[\Psi_2(k)] \leq \mathcal{O}\left(\gamma^k\right) \quad \text{with} \quad \gamma = \rho\left(\mathbb{E}\left[\mathbf{K} \otimes \mathbf{K}\right] \left(\mathbf{J}^\perp \otimes \mathbf{J}^\perp\right)\right) < 1$$



$\|\mathbf{x}(\tau_n) - x_{\text{ave}}\mathbf{1}\|^2 < C\gamma^{\tau_n}$ $\quad$ $\|\mathbf{x}(\tau_{n+1}) - x_{\text{ave}}\mathbf{1}\|^2 < C\gamma^{\tau_{n+1}}$

$\max |x_i(k) - x_{\text{ave}}|^2$ is non-increasing

Example : Broadcast sum-Weight (BW) Gossip

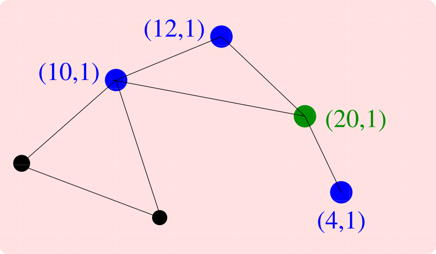


At time k, let i be the activate node [Iutzeler2013]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip

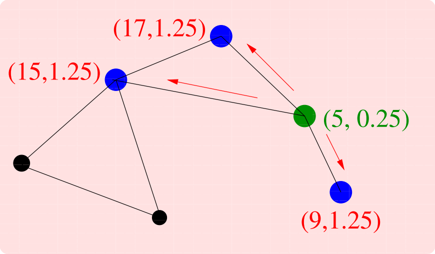


At time k, let i be the activate node [Iutzeler2013]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip

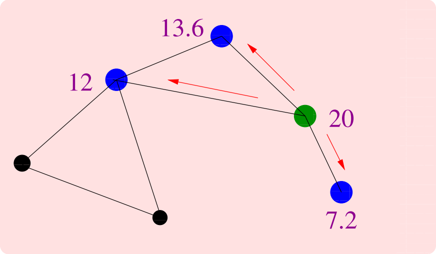


At time k, let i be the activate node [Iutzeler2013]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip



At time k, let i be the activate node [Iutzeler2013]

- Node i updates :
$$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates :
$$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Example : Broadcast sum-Weight (BW) Gossip

$$\mathbf{K}_i = \begin{bmatrix} \ddots 1 \ddots & & \frac{1}{d_i+1} & \\ & \ddots 1 \ddots & & \\ & & \frac{1}{d_i+1} & \\ & & & \ddots 1 \ddots \end{bmatrix}$$

At time k, let i be the activate node [Iutzeler2013]

- Node i updates : $$\begin{cases} s_i(k+1) = \frac{s_i(k)}{d_i+1} \\ w_i(k+1) = \frac{w_i(k)}{d_i+1} \end{cases}$$
- i broadcasts $\left(\frac{s_i(k)}{d_i+1}; \frac{w_i(k)}{d_i+1}\right)$ to all its neighbors
- Each neighbor $j \in \mathcal{N}_i$ updates : $$\begin{cases} s_j(k+1) = s_j(k) + \frac{s_i(k)}{d_i+1} \\ w_j(k+1) = w_j(k) + \frac{w_i(k)}{d_i+1} \end{cases}$$

- Take into account broadcast nature of wireless channel
- No row-stochastic matrices but converges to the true value

Back to [Boyd2006]

Remark

If update matrices are doubly-stochastic, then

$$w_i(k) = 1 \quad \Leftrightarrow \quad \Psi_1(k) = 1$$

and

$$s_i(k) = x_i(k), \quad \forall i, k$$

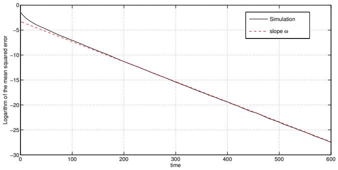
Our analysis still holds for standard single-variate algorithms

- In [Boyd2006], exponential convergence with $\rho(\mathbb{E}[\mathbf{K}\mathbf{K}^{\mathrm{T}}] - \mathbf{J})$
- In [Iutzeler2013], exponential convergence with $\rho(\mathbb{E}[\mathbf{K} \otimes \mathbf{K}](\mathbf{J}^{\perp} \otimes \mathbf{J}^{\perp}))$

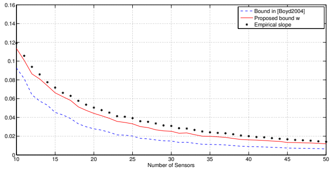
Part 4 : Numerical illustrations

Tightness of the analysis

- Random Geometric Graphs with radius of order $\sqrt{\log(N)/N}$
- Unless otherwise staded, $N = 100$



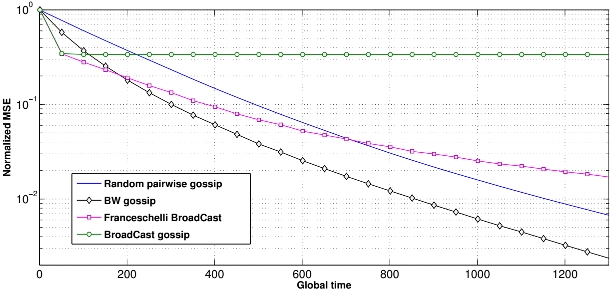
MSE for BW Gossip vs. time



Slope for Random Gossip vs. N

The proposed bound is very tight

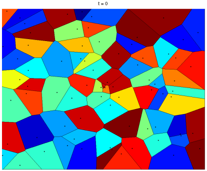
Comparison with existing methods



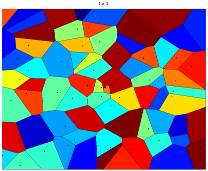
MSE for various averaging algorithms

Proposed algorithm outperforms existing ones

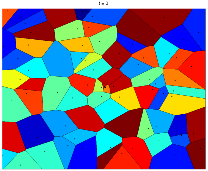
A funny example



RANDOM GOSSIP



BROADCAST GOSSIP



BW GOSSIP

Min Max Mean

Part 5 : Perspectives

Distributed optimization

$$\hat{x}_{\rm opt} = \arg\min_x f(x) \triangleq \sum_{i=1}^{N} f_i(x)$$

with

- f_i convex and known only at node i
- f unknown anywhere
- ... so only local exchanges allowed

Most standard way to fix this problem : distributed gradient algorithm.

Gradient step : $\tilde{x}_i(k+1) = x_i(k) - \gamma_k f_i'(x_i(k))$ for each node i.

Gossip step : $\mathbf{x}(k+1) = \mathbf{K}\tilde{\mathbf{x}}(k+1)$ with $\mathbf{K}$ used for average computation

▶ **Distributed average computation is needed**

- Sum-Weight based distributed optimization : [Nedich2013] for sync. case
- **Applications :** machine learning, big data, cloud computing, ad hoc networks, ...

References

[DeGroot1974] M. DeGroot, “Reaching a consensus,” Jnl of ASA, 1974.

[Chatterjee1977] S. Chatterjee et al., “Towards consensus : some convergence theorems on repeating averaging,” Jnl of Applied Proba, 1977.

[Kempe2003] D. Kempe et al., “Gossip-based computation of aggregate infomration,” IEEE FOCS, 2003.

[Boyd2006] S. Boyd et al., “Randomized Gossip Algorithms,” IEEE TIT, 2006.

[Tahbaz2008] A. Tahbza-Salehi et al., “A necessary and sufficient condition for consensus over random networks,” IEEE TAC, 2008.

[Aysal2009] T.C. Aysal et al., “Broadcast Gossip Algorithms for Consensus,” IEEE TSP, 2009.

[Benezit2010] F. Bénézit et al., “Weighted Gossip : distributed averaging using non-doubly-stochastic matrices ,” IEEE ISIT, 2010.

[Iutzeler2013] F. Iutzeler et al., “Analysis of Sum-Weight-like algorithms for averaging in WSN,” IEEE TSP, 2013.

[Nedich2013] A. Nedich et al., “Distributed optimization over time-varying directed graphs,” IEEE CDC, 2013.