What are the Optimal Communication Weights for Decentralized Optimization ?

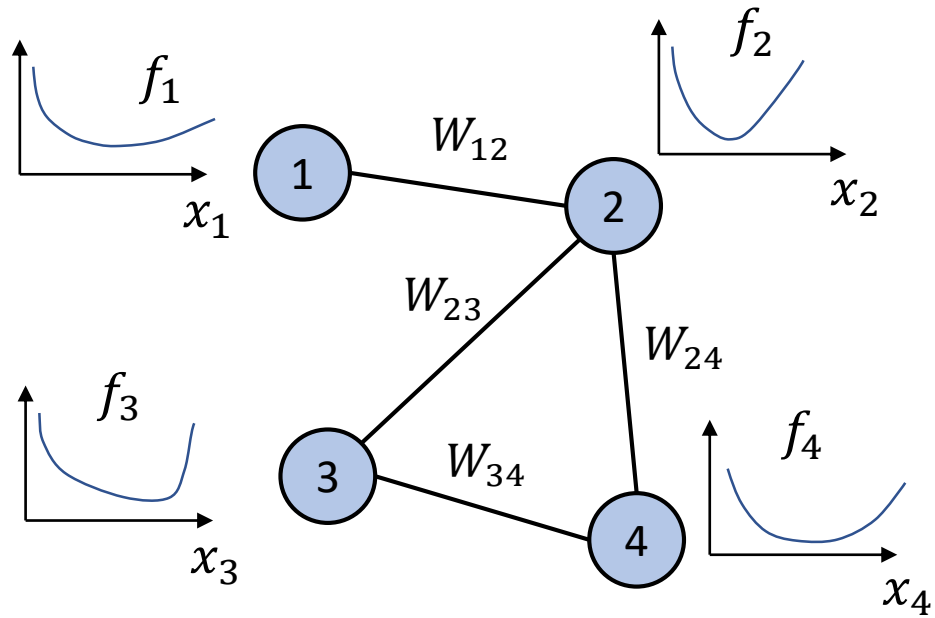
Sébastien Colla, Julien Hendrickx

Mathematical Engineering Department,
UCLouvain (Belgium)

Benelux Meeting 2024

Decentralized Optimization

$$\min_x f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$



Decentralization

- Local function: f_i

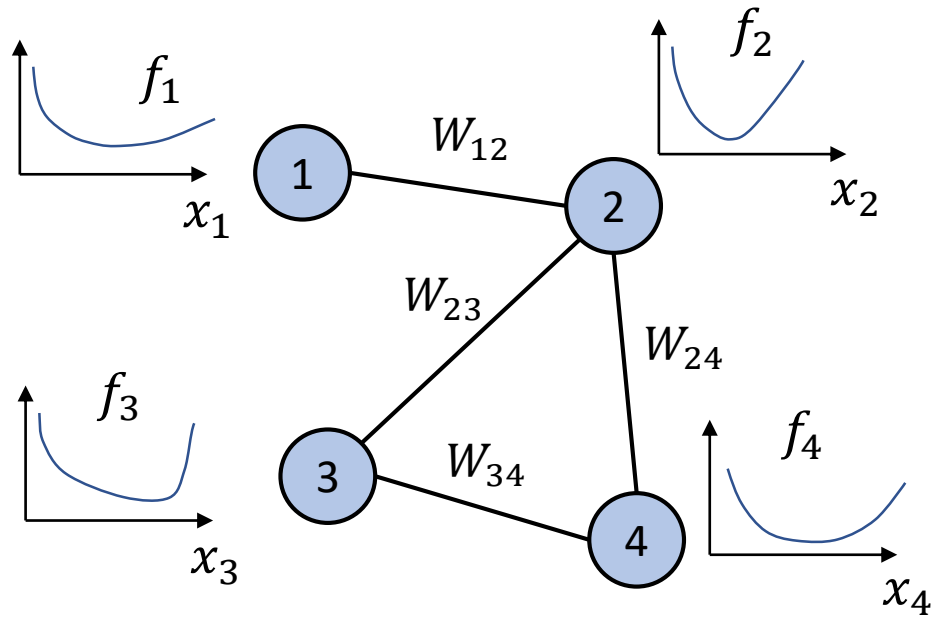
Decentralized Optimization

$$\min_{x} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

$$\Updownarrow$$

$$\min_{x_1,\dots,x_n} F_s(x_1,\dots,x_n) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_i)$$

$$\text{s.t. } x_i = x_j \quad \forall\,(i,j) \text{ neighbors}$$



Decentralization

- Local function: f_i
- Local copy of x: x_i

Iterative algorithm

- Local computations
- Local communications (W)
  so that $x_i = x_j$ (eventually)

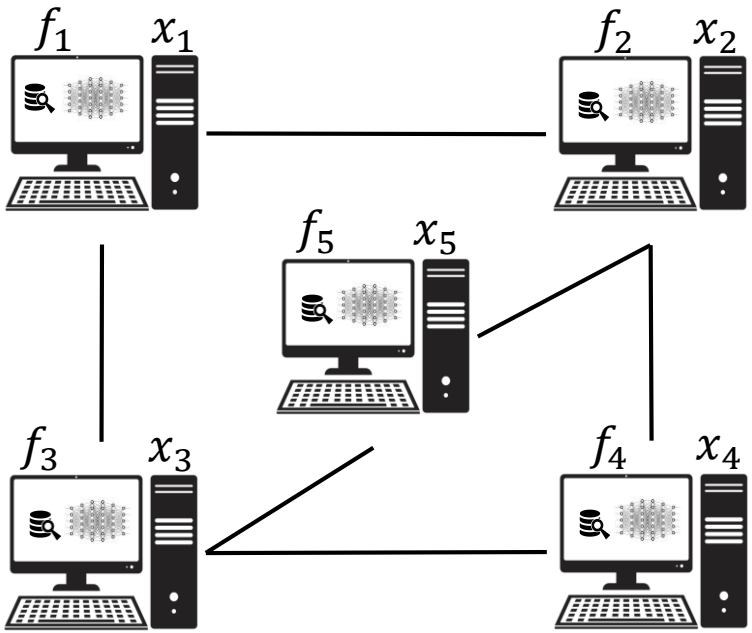
Motivations: Decentralized Machine Learning

Notations

- Model parameters x
- Data set $\{d \in \mathcal{D}\}$

Model training

$$\min_{x} \sum_{d \in D} \mathrm{Error}(x, d)$$



Decentralization

- Part of the data $\mathcal{D}_i$
- Local function

$$f_i(x) = \sum_{d \in \mathcal{D}_i} \mathrm{Error}(x, d)$$

- Local copy of x

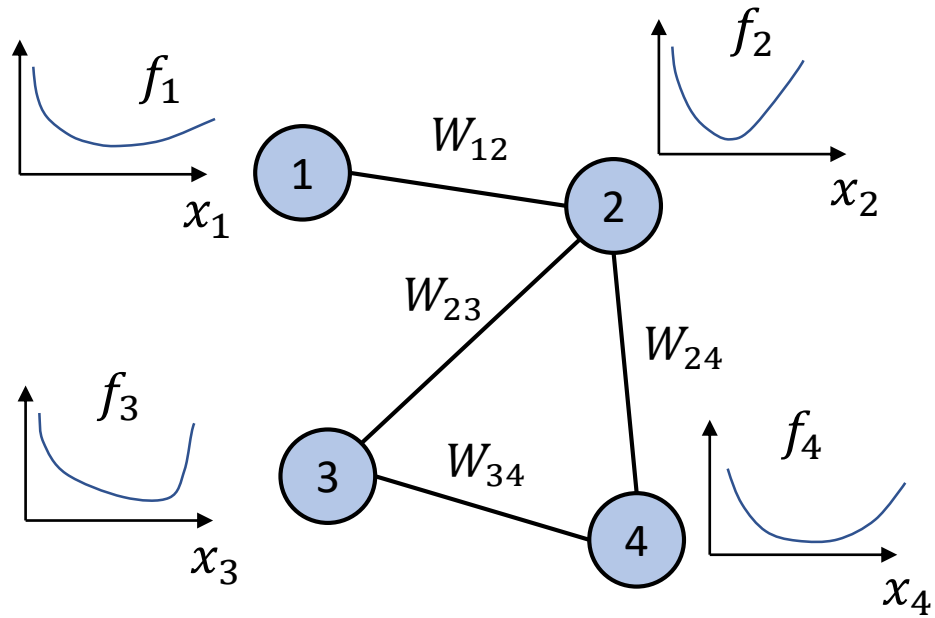
Motivations Big data – Privacy – Speed Up

DIGing Algorithm

Problem

$$\min_{x_1,\dots,x_n} F_s(x_1,\dots,x_n) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_i)$$

$$\text{s.t. } x_i = x_j \quad \forall\,(i,j) \text{ neighbors}$$



DIGing Algorithm

For each iteration k and each agent i:

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k \qquad \textit{Local gradient step}$$

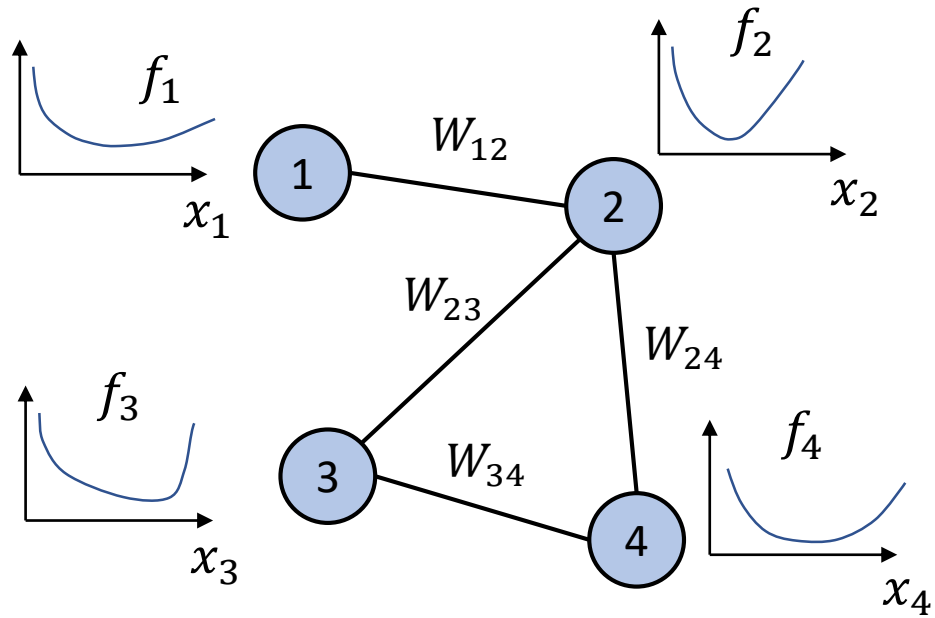
$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i\left(x_i^k\right) \qquad \textit{Gradient tracking}$$

DIGing Algorithm

Problem

$$\min_{x_1,\dots,x_n} F_s(x_1,\dots,x_n) = \frac{1}{n}\sum_{i=1}^{n} f_i(x_i)$$

$$\text{s.t. } x_i = x_j \quad \forall\,(i,j) \text{ neighbors}$$



DIGing Algorithm

For each iteration k and each agent i:

estimates $\frac{1}{n}\sum_{j=1}^{n}\nabla f_j(x_j^k)$

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k \qquad \textit{Local gradient step}$$

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i\left(x_i^k\right) \qquad \textit{Gradient tracking}$$

DIGing Algorithm

For each iteration k and each agent i:

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k$$ *Local gradient step*

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i(x_i^k)$$ *Gradient tracking*

Convergence Guarantee

For any local functions $f_i \in \mathcal{F}_{\mu,L}$

For any starting point such that $\frac{1}{n}\sum_{i=1}^{n} \left\| x_i^0 - x^* \right\|^2 \leq R^2$

For any mixing matrices $W \in \left\{ W: \begin{array}{l} \text{symmetric} \\ \text{doubly stochastic } (\lambda_1 = 1) \\ \max\{|\lambda_2(W)|, |\lambda_n(W)|\} \leq \lambda \end{array} \right\}$

DIGing Algorithm

For each iteration k and each agent i:

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k$$ *Local gradient step*

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i(x_i^k)$$ *Gradient tracking*

Convergence Guarantee

For any local functions $f_i \in \mathcal{F}_{\mu,L}$

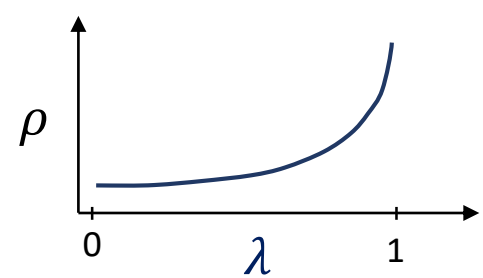
For any starting point such that $\frac{1}{n}\sum_{i=1}^{n} \left\|x_i^0 - x^*\right\|^2 \le R^2$

For any mixing matrices $W \in \left\{ W: \begin{array}{l} \text{symmetric} \\ \text{doubly stochastic } (\lambda_1 = 1) \\ \max\{|\lambda_2(W)|, |\lambda_n(W)|\} \le \lambda \end{array} \right\}$

We have

$$\frac{1}{n}\sum_{i=1}^{n} \left\|x_i^t - x^*\right\|^2 \le C\, \rho^t(\mu, L, R, \alpha, \lambda)$$

convergence rate $\rho \in (0,1)$



DIGing Algorithm

For each iteration k and each agent i:

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k$$ *Local gradient step*

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i(x_i^k)$$ *Gradient tracking*

Convergence Guarantee

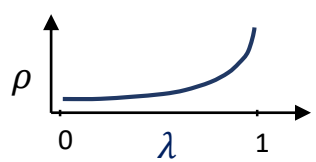
For any local functions $f_i \in \mathcal{F}_{\mu,L}$

For any starting point such that $\frac{1}{n}\sum_{i=1}^{n} \left\|x_i^0 - x^*\right\|^2 \leq R^2$

For any mixing matrices $W \in \left\{ W: \begin{array}{l} \text{symmetric} \\ \text{doubly stochastic } (\lambda_1 = 1) \\ \max\{|\lambda_2(W)|, |\lambda_n(W)|\} \leq \lambda \end{array} \right\}$

We have

$$\frac{1}{n}\sum_{i=1}^{n} \left\|x_i^t - x^*\right\|^2 \leq C\, \rho^t(\mu, L, R, \alpha, \lambda)$$



Minimize con. rate ρ
- tune α
- tune W (and λ)

DIGing Algorithm

For each iteration k and each agent i:

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k$$ *Local gradient step*

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i(x_i^k)$$ *Gradient tracking*

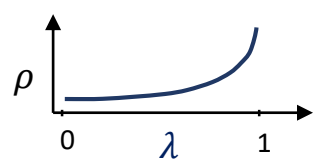
NATURAL QUESTIONS

- How to chose the mixing matrix W?
- Do weights minimizing λ give the best performance?

$$W \in \left\{ W: \begin{array}{l} \text{symmetric} \\ \text{doubly stochastic } (\lambda_1 = 1) \\ \max\{|\lambda_2(W)|, |\lambda_n(W)|\} \leq \lambda \end{array} \right\}$$

We have

$$\frac{1}{n}\sum_{i=1}^{n} \left\| x_i^t - x^* \right\|^2 \leq C\, \rho^t(\mu, L, R, \alpha, \lambda)$$



Minimize con. rate ρ

- tune α
- tune W (and λ)

DIGing Algorithm

For each iteration k and each agent i:

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k$$

Local gradient step

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i(x_i^k)$$

Gradient tracking

NATURAL QUESTIONS

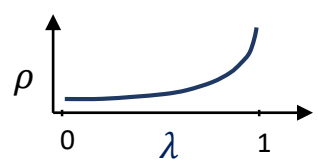
- How to chose the mixing matrix W?
- Do weights minimizing λ give the best performance?

SPOILER: NO

$$W \in \left\{ W: \begin{array}{l} \text{symmetric} \\ \text{doubly stochastic } (\lambda_1 = 1) \\ \max\{|\lambda_2(W)|, |\lambda_n(W)|\} \leq \lambda \end{array} \right\}$$

We have

$$\frac{1}{n}\sum_{i=1}^{n} \left\| x_i^t - x^* \right\|^2 \leq C\, \rho^t(\mu, L, R, \alpha, \lambda)$$



Minimize con. rate ρ

- tune α
- tune W (and λ)

Communication Weights

The averaging matrix $W \in \mathbb{R}^{n \times n}$ satisfies

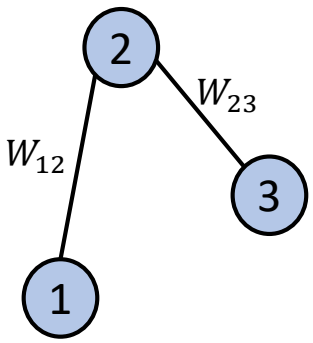
(1) $W^T = W,$ (Symmetry)

(2) $W1 = 1,$ $(\text{and } 1^T W = 1^T)$ (Averaging Consensus)

(3) $W \in T,$ (Topology)

where $T = \{ W : W_{ij} = 0 \text{ if no edge between } i \text{ and } j \}$.

Degree of freedom: *one weight for each edge*



$$W = \begin{bmatrix} 1 - W_{12} & W_{12} & 0 \\ W_{12} & 1 - W_{12} - W_{23} & W_{23} \\ 0 & W_{23} & 1 - W_{23} \end{bmatrix}$$

Communication Weights

The averaging matrix $W \in \mathbb{R}^{n \times n}$ satisfies

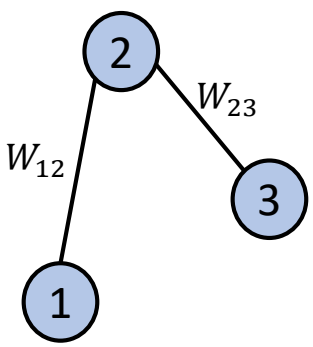
(1) $W^T = W$, (Symmetry)

(2) $W1 = 1$, (and $1^T W = 1^T$) (Averaging Consensus)

(3) $W \in T$, (Topology)

where $T = \{ W : W_{ij} = 0 \text{ if no edge between } i \text{ and } j \}$.

Degree of freedom: *one weight for each edge*



$$W = I - B\mathrm{diag}(w)B^T,$$

where
- I: identity matrix
- B: graph incidence matrix
- w: vector of weights

Satisfies (1)-(3) for any w

$$W = \begin{bmatrix} 1 - W_{12} & W_{12} & 0 \\ W_{12} & 1 - W_{12} - W_{23} & W_{23} \\ 0 & W_{23} & 1 - W_{23} \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix} \qquad \mathrm{diag}(w) = \begin{bmatrix} W_{12} & 0 \\ 0 & W_{23} \end{bmatrix}$$

Weights Heuristics

$$W = I - B\mathrm{diag}(w)B^T,$$

- Minimum-λ weights

$$w_{\lambda^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \left\| I - B\mathrm{diag}(w)B^T - \frac{11^T}{n} \right\|_2$$

Second Largest Eigenvalue Modulus (SLEM)

$$\boxed{\begin{aligned} \lambda &= \max\{|\lambda_2(W)|, |\lambda_n(W)|\} \\ &= \left\| W - \frac{11^T}{n} \right\|_2 \end{aligned}}$$

➡ **Optimal weights** for the pure linear consensus [Xiao, Boyd 2004]

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k$$

➡ Classical choice in decentralized optimization literature
But requires global knowledge of the network

Weights Heuristics

$$W = I - B\mathrm{diag}(w)B^T,$$

- Minimum-λ weights

$$w_{\lambda^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \; \lambda = \left\| I - B\mathrm{diag}(w)B^T - \frac{11^T}{n} \right\|_2$$

⇨ *optimal for pure consensus [Xiao, Boyd 2004]*
global knowledge needed

- Metropolis weights

$$w_e = \frac{1}{\max\{\deg_i, \deg_j\} + 1}$$ for each edge $e = (i, j)$

➡ *only local knowledge (degrees) needed*

- Lazy Metropolis weights

$$w_e = \frac{1/2}{\max\{\deg_i, \deg_j\} + 1}$$ for each edge $e = (i, j)$

Weights Heuristics

$$W = I - B\mathrm{diag}(w)B^T,$$

Minimum nuclear norm weights

$$w_{\lambda^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad \|W\|_\Sigma = \sum_i |\lambda_i(W)|$$

Minimum total effective resistance weights [Gosh 2008]

$$w_{R^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad R_{\mathrm{tot}}(W) = n \sum_{i=2}^{n} \frac{1}{1 - \lambda_i(W)}$$

Least mean-square deviation weights [Xiao 2007]

$$w_{\delta^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad \sum_{i=2}^{n} \frac{1}{1 - \lambda_i^2(W)}$$

Weights Heuristics

$$W = I - B\mathrm{diag}(w)B^T,$$

Minimum nuclear norm weights

$$w_{\lambda^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad \|W\|_\Sigma = \sum_i |\lambda_i(W)|$$

Average commute time in Markov Chain

Minimum total effective resistance weights [Gosh 2008]

$$w_{R^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad R_{\mathrm{tot}}(W) = n \sum_{i=2}^{n} \frac{1}{1 - \lambda_i(W)}$$

Least mean-square deviation weights [Xiao 2007]

$$w_{\delta^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad \sum_{i=2}^{n} \frac{1}{1 - \lambda_i^2(W)}$$

Weights Heuristics

$$W = I - B\mathrm{diag}(w)B^T,$$

Minimum nuclear norm weights

$$w_{\lambda^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad \|W\|_\Sigma = \sum_i |\lambda_i(W)|$$

Average commute time in Markov Chain

Minimum total effective resistance weights [Gosh 2008]

$$w_{R^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad R_{\mathrm{tot}}(W) = n \sum_{i=2}^{n} \frac{1}{1-\lambda_i(W)}$$

Least mean-square deviation weights [Xiao 2007]

$$w_{\delta^*} = \underset{w \in \mathbb{R}^m}{\mathrm{argmin}} \quad \sum_{i=2}^{n} \frac{1}{1-\lambda_i^2(W)} \quad = \lim_{k\to\infty} \mathbb{E} \sum_{i=1}^{n} \left(x_i^k - \bar{x}\right)^2$$

Optimal weights for consensus with additive noise [Xiao 2007]

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k + v_i^k$$

Worst-case performance evaluation

$$Err(W,\alpha) = \max_{x^*, x_i^k, s_i^k, f_i} Perf(f_i, x_i^0, \dots, x_i^K, x^*)$$

(for $\mathrm{i} = 1, \dots, \mathrm{n}$)

such that

$$x_i^{k+1} = \sum_j W_{ij}\, x_j^k - \alpha s_i^k$$

$$s_i^{k+1} = \sum_j W_{ij}\, s_j^k + \nabla f_i\left(x_i^{k+1}\right) - \nabla f_i(x_i^k)$$

Algorithm (DIGing)

$$f_i \in \mathcal{F}_{\mu,L}$$

Class of functions

x_i^0, s_i^0 satisfy initial conditions

$$\frac{1}{n}\sum_{i=1}^{n} f_i(x^*) = 0$$

Optimal condition

Performance Estimation Problem (PEP)

➡ Solved exactly with SDP reformulation

PESTO toolbox

Optimal Weights

$$(w^*, \alpha^*) = \underset{w \in \mathbb{R}^m,\ \alpha \geq 0}{\operatorname{argmin}} \quad Err(I - B\operatorname{diag}(w)B^T, \alpha)$$

computed with PEP
via PESTO toolbox

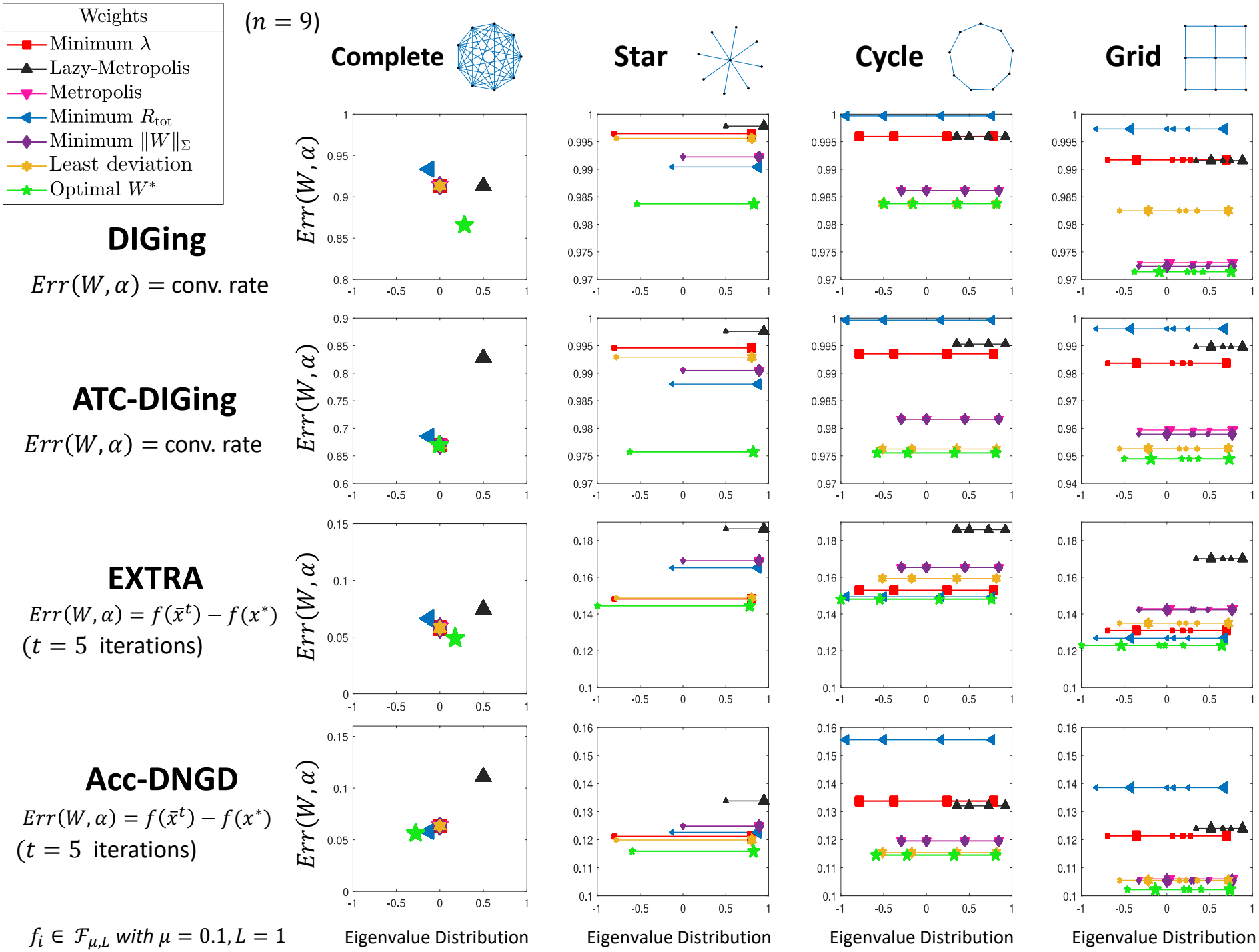
solved with a *zero-order* method
(pattern search)

$$W^* = I - B\operatorname{diag}(w^*)B^T$$

Symmetry, $(W^*)^T = W^*$

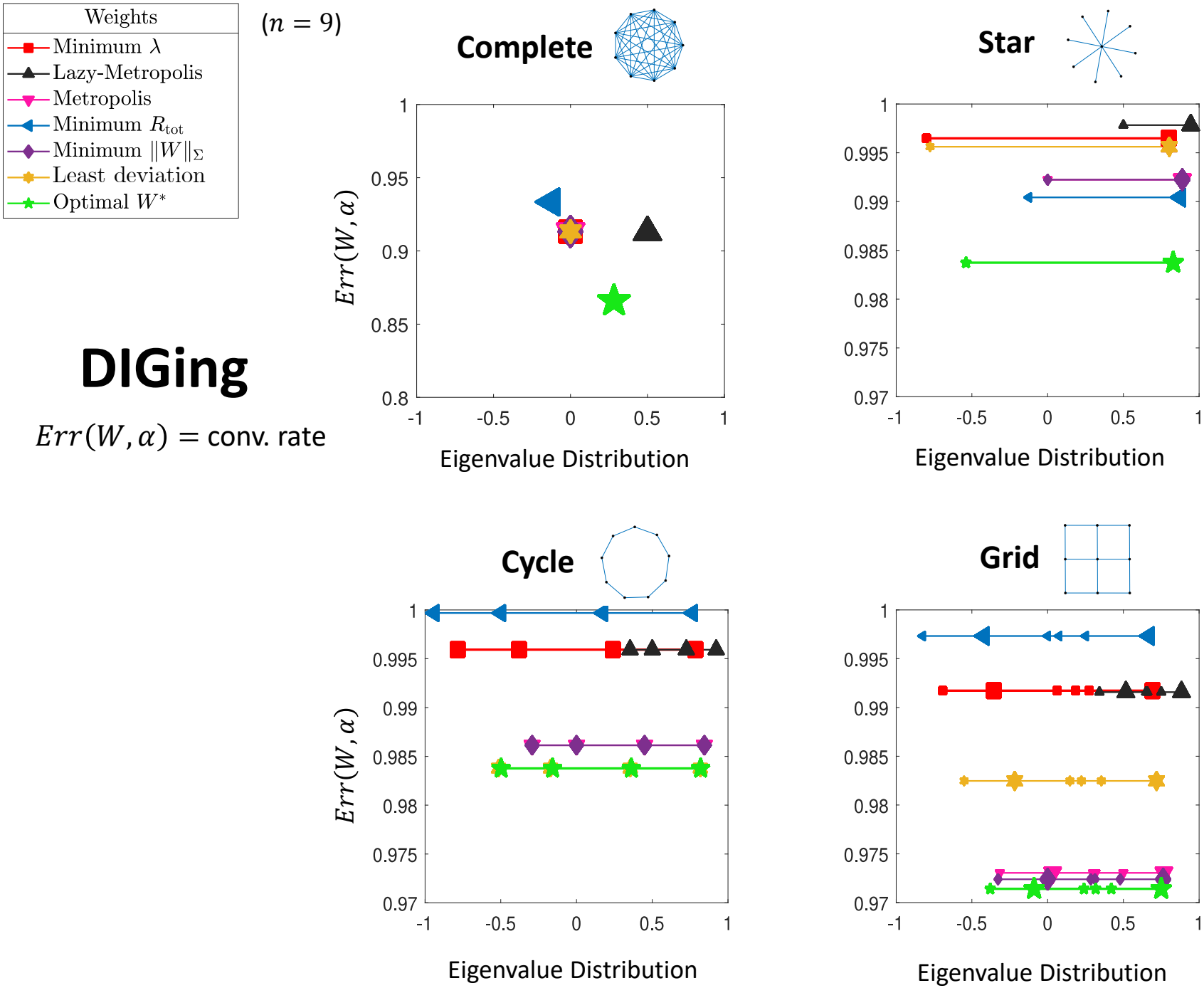
Average consensus, $W^*1 = 1$

Topology $W^* \in T$


Weights
Minimum λ
Lazy-Metropolis
Metropolis
Minimum R_{tot}
Minimum $\|W\|_{\Sigma}$
Least deviation
Optimal W^*
$(n = 9)$
Complete
Star
Cycle
Grid
DIGing
$Err(W, \alpha)$ = conv. rate
ATC-DIGing
$Err(W, \alpha)$ = conv. rate
EXTRA
$Err(W, \alpha) = f(\bar{x}^t) - f(x^*)$
$(t = 5$ iterations)
Acc-DNGD
$Err(W, \alpha) = f(\bar{x}^t) - f(x^*)$
$(t = 5$ iterations)
$Err(W, \alpha)$
Eigenvalue Distribution
$f_i \in \mathcal{F}_{\mu,L}$ with $\mu = 0.1, L = 1$

DIGing

$Err(W, \alpha)$ = conv. rate



$f_i \in \mathcal{F}_{\mu,L}$ *with* $\mu = 0.1, L = 1$

Conclusion | On the Optimal Communication Weights for Decentralized Optimization

Using PEP framework

- **Compute optimal weights** over an undirected network
- **Evaluate different weights** heuristics

Conclusion | On the Optimal Communication Weights for Decentralized Optimization

Using PEP framework

- **Compute optimal weights** over an undirected network
- **Evaluate different weights** heuristics

Outcomes

- Weights minimizing λ (SLEM) are **suboptimal**
  - *Not the best determinant for the performance of a weight matrix in decentralized optimization*
- A good characterization of weights performance should take into account the distribution of **all eigenvalues**.
  - *Not found yet… But the weights optimal for consensus with noise [Xiao 2007] seem to perform well.*

Conclusion | On the Optimal Communication Weights for Decentralized Optimization

Using PEP framework

- **Compute optimal weights** over an undirected network
- **Evaluate different weights** heuristics

Outcomes

- Weights minimizing λ (SLEM) are **suboptimal**
  - *Not the best determinant for the performance of a weight matrix in decentralized optimization*
- A good characterization of weights performance should take into account the distribution of **all eigenvalues**.
  - *Not found yet… But the weights optimal for consensus with noise [Xiao 2007] seem to perform well.*

Related paper : S. Colla, J. M. Hendrickx, "*On the Optimal Communication Weights in Distributed Optimization Algorithms*", submitted to MTNS 2024.

References

Sébastien Colla

[Colla 2024] S. Colla, J. M. Hendrickx, “On the Optimal Communication Weights in Distributed Optimization Algorithms”, submitted to MTNS 2024.

[Xiao 2004] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging”, Systems & Control Letters, 53(1), (2004).

[Xiao 2007] L. Xiao, S. Boyd and S.J. Kim, “Distributed average consensus with least-mean-square deviation”, J. of parallel and distributed computing, (2007).

[Gosh 2008] A. Ghosh, S. Boyd, & A. Saberi, “Minimizing effective resistance of a graph”, SIAM review, 50(1), 37-66, (2008).