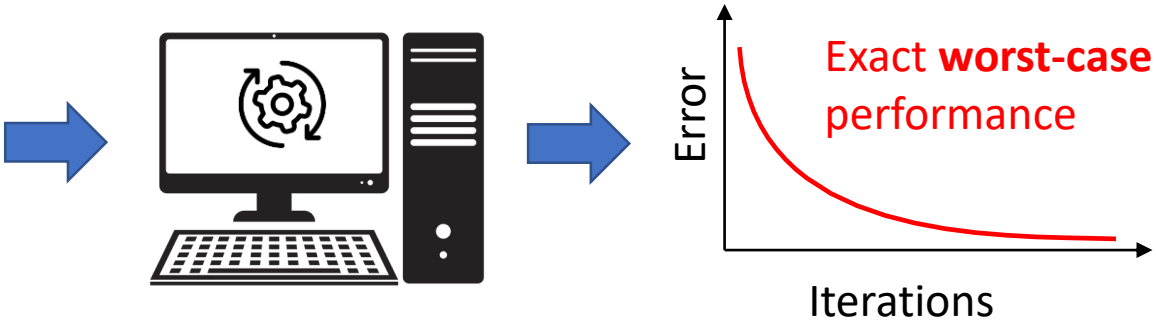
Performance Estimation Problem For Decentralized Optimization Methods

Sébastien Colla, Julien Hendrickx

Benelux Meeting – 6th July 2022

```
function MyDecentralizedAlgo()
    N = 10;         % number of agents
    x0 = init(N);   % initial points
    x = x0;

    for k=1:niter
        % any local computations
        % any local communications
        x = update(x,N);
    end
end
```

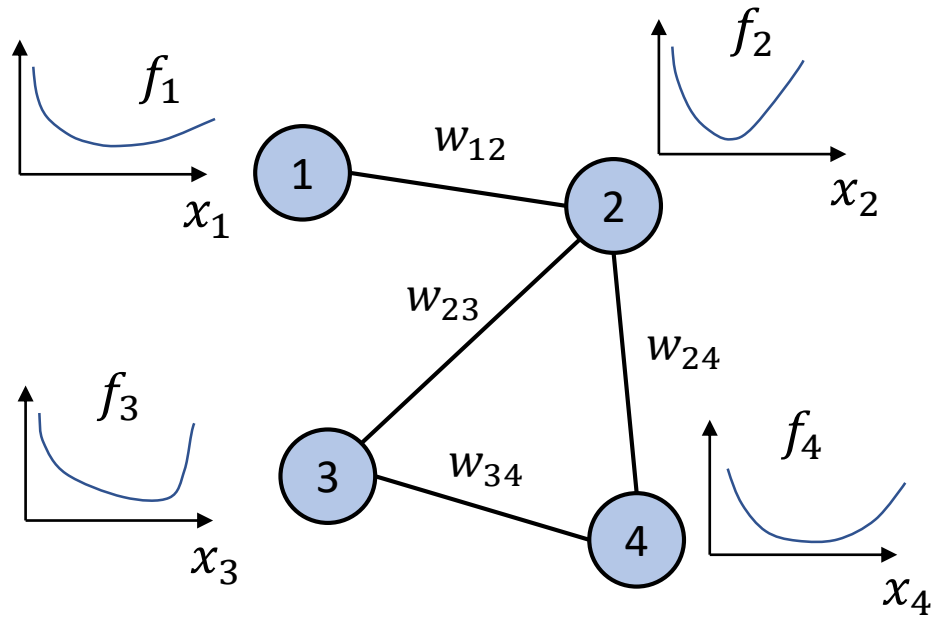


Decentralized Optimization

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

$$\Updownarrow$$

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^{N} f_i(x_i)$$
$$\text{s.t. } x_i = x_j \quad \forall\, (i,j) \text{ neighbors}$$



Decentralization

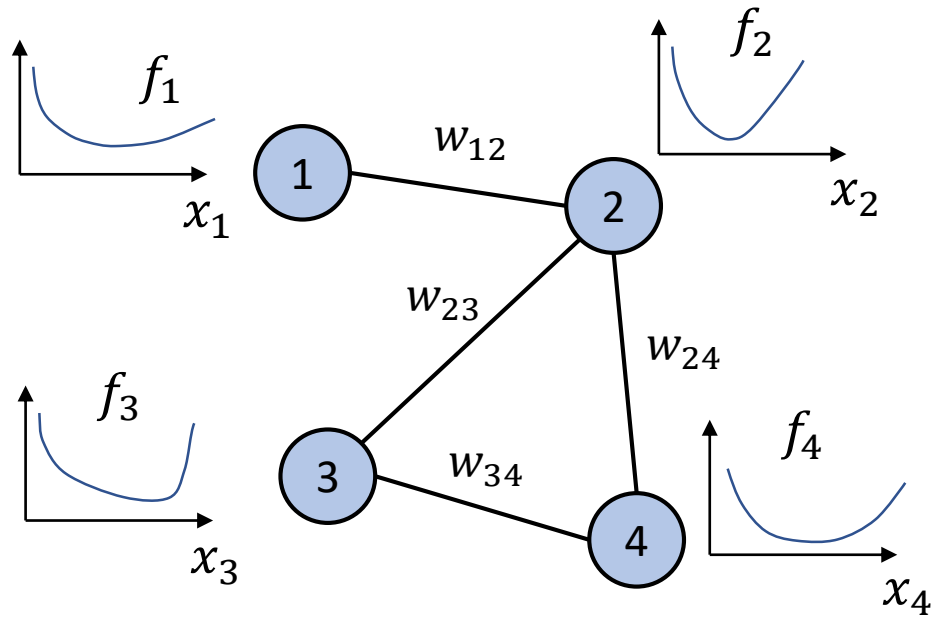
- Local function: f_i
- Local copy of x: x_i

Iterative algorithm

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

Decentralized Gradient Descent (DGD)

$$\min_{x_1,\ldots,x_N} \sum_i f_i(x_i)$$
$$\text{s.t. } x_i = x_j \quad \forall\, (i,j) \text{ neighbors}$$



Decentralization

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

Decentralized Gradient Descent (DGD)

For each iteration k

$$y_i^k = \sum_j w_{ij}\, x_j^k \qquad \textit{Consensus step}$$

$$x_i^{k+1} = y_i^k - \alpha \nabla f_i(x_i^k) \qquad \textit{Local gradient step}$$

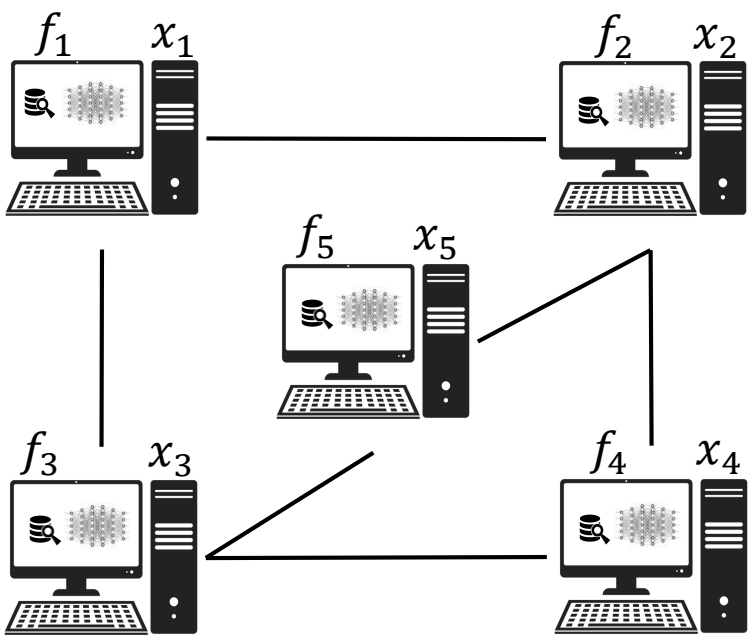
Motivations: Decentralized Machine Learning

Notations

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

Model training

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



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

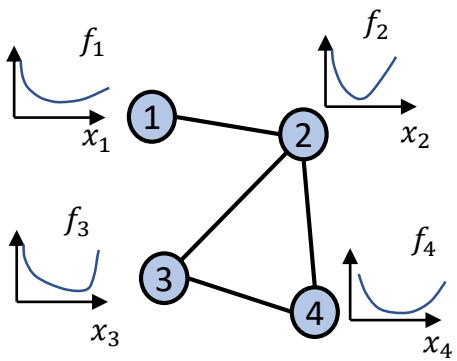
Motivations Big data – Privacy – Speed Up

Decentralized Optimization

Many challenges for better methods

BUT

Analysis highly complex



optimization

communication

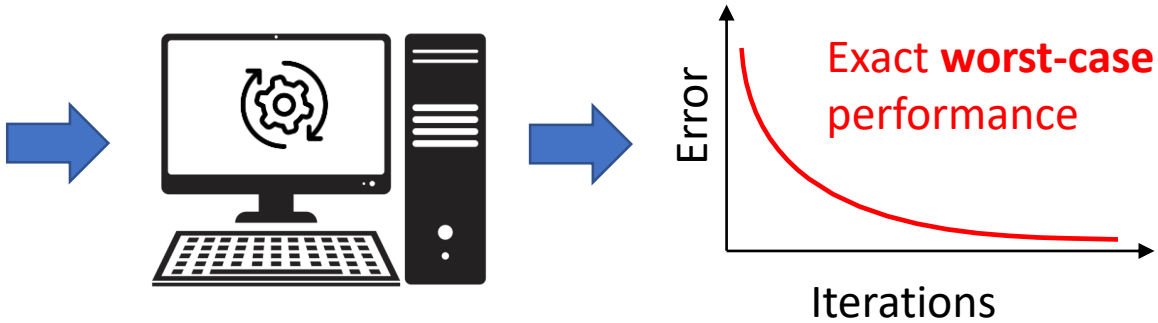
- Performance bounds: complex and **conservative**
- Difficult algorithms comparisons
- Difficult parameters tuning

Objective

```
function MyDecentralizedAlgo()
    N = 10;        % number of agents
    x0 = init(N);  % initial points
    x = x0;

    for k=1:niter
        % any local computations
        % any local communications
        x = update(x,N);
    end
end
```

For convex functions



Impact for decentralized optimization

- Access to **accurate performance** of methods
- Easier **comparison and tuning** of algorithms
- **Rapid exploration** of new algorithms.

Performance Estimation Problem (PEP)

Idea: Worst-cases are solutions to optimization problems

$$\max_{f,\, x^0, \dots, x^K} \quad \mathrm{perf}(f, x^0, \dots, x^K) \overset{e.g.}{=} f(x^K) - f(x^*)$$

With	f	$\in$	class of functions	*e.g.* $\mathcal{F}_{\mu,L}$
	x_0		initial condition	*e.g.* $\|x^0 - x^*\|^2 \leq 1$
	x^k		from the algorithm analyzed	

Infinite-Dimensional problem

Via
- *Discretization*
- *SDP reformulation*

PEP can be solved exactly for a wide class of **centralized** first-order algorithms.

Existing toolbox (PESTO)

[Taylor17]

Performance Estimation Problem (PEP)

Idea: Worst-cases are solutions to optimization problems

$$\max_{\substack{f_i,\, x_i^0,\ldots,x_i^K,\, W \\ (i=1,\ldots,N)}} \quad \mathrm{perf}(f_i,\, x_i^0,\, \ldots,\, x_i^K)$$

With	f_i	$\in$	class of functions
	x_i^0		initial condition
	x_i^k		from the algorithm analyzed
	W	$\in$	class of network matrices

How to represent a class of ***communication network matrices*** *?*

PEP for DGD: network given a priori

$$\max_{\substack{f_i,\, x_i^0,\dots,x_i^K,\, W \\ y_i^0,\dots,y_i^{K-1}}} \quad \mathrm{perf}(f_i,\, x_i^0,\dots,x_i^K)$$

With $f_i \in$ class of functions

x_i^0 initial condition

W **given network matrix**

Iterates from DGD
$$\begin{cases} y_i^k = \sum_j w_{ij}\, x_j^k \\ x_i^{k+1} = y_i^k - \alpha \nabla f_i(x_i^k) \end{cases}$$

For all $i = 1 \dots N$,
For all $k = 0 \dots K-1$

Exact Formulation

PEP for DGD: class of networks

$$\max_{\substack{f_i, x_i^0, \dots, x_i^K, W \\ y_i^0, \dots, y_i^{K-1}}} \quad \mathrm{perf}(f_i, x_i^0, \dots, x_i^K)$$

With $f_i \in$ class of functions

x_i^0 initial condition

$\boldsymbol{W}$ **Any symmetric doubly stochastic matrix with given range of eigenvalues $[\lambda^-, \lambda^+]$**

Find constraints between y_i^k and x_i^k

Iterates from DGD

$$\begin{cases} \boxed{y_i^k = \sum_j w_{ij}^{?} x_j^k} \\ x_i^{k+1} = y_i^k - \alpha \nabla f_i(x_i^k) \end{cases}$$

For all $i = 1 \dots N$,
For all $k = 0 \dots K-1$

PEP for DGD: class of networks

$$\max_{\substack{f_i,\, x_i^0,\dots,x_i^K,\cancel{W} \\ y_i^0,\dots,y_i^{K-1}}} \quad \mathrm{perf}(f_i,\, x_i^0,\dots,x_i^K)$$

With $f_i \in$ class of functions

x_i^0 initial condition

$\boldsymbol{W}$ ~~**Any symmetric doubly stochastic matrix with given range of eigenvalues** $[\boldsymbol{\lambda}^-, \boldsymbol{\lambda}^+]$~~

Find constraints between y_i^k and x_i^k

Iterates from DGD

$$\begin{cases} \boxed{\cancel{y_i^k = \sum_j w_{ij}\, x_j^k}} \\ x_i^{k+1} = y_i^k - \alpha \nabla f_i(x_i^k) \end{cases}$$

For all $i = 1 \dots N$,
For all $k = 0 \dots K-1$

PEP for DGD: Spectral formulation

(Relaxation)

$$\max_{\substack{f_i,\, x^0,\dots,x^K \\ y^0,\dots,y^{K-1}}} \quad \mathrm{perf}(f_i,\, x^0,\dots,x^K)$$

With $f_i \in$ class of functions

x_i^0 initial condition

Iterates from DGD $\left\{ x_i^{k+1} = y_i^k - \alpha \nabla f_i(x_i^k) \right.$ For all $i = 1 \dots N$, For all $k = 0 \dots K-1$

Consensus steps

$Y = WX$

W symmetric doubly stochastic $\lambda(W) \in [\lambda^-, \lambda^+]$

Necessary Constraints

$$\bar{X} = \bar{Y}$$

$$\lambda^- X_\perp^T X_\perp \preccurlyeq X_\perp^T Y_\perp \preccurlyeq \lambda^+ X_\perp^T X_\perp$$

$$(Y_\perp - \lambda^- X_\perp)^T (Y_\perp - \lambda^+ X_\perp) \preccurlyeq 0$$

Notations

$X_{ik} = x_i^k,\ Y_{ik} = y_i^k$

$\bar{X}, \bar{Y}$: agents average

$X_\perp, Y_\perp$: centered matrices

Upper bounds for the worst-case performance of DGD

Our tool for automatic performance estimation

Apply to any decentralized method using consensus $y = Wx$

- Exact formulation

  exact worst-case performance,
  specific to a given matrix W

  Available in PESTO

- Spectral formulation

  upper bound on worst-case performance,
  valid for an entire **spectral class** of network matrices

  Available in PESTO

Note: In both cases, the size of the PEP problem **depends** on the number of iterations **K** and the number of agents **N.**

Results of PEP for DGD

Problem $\min_x f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$ with optimal solution x^*

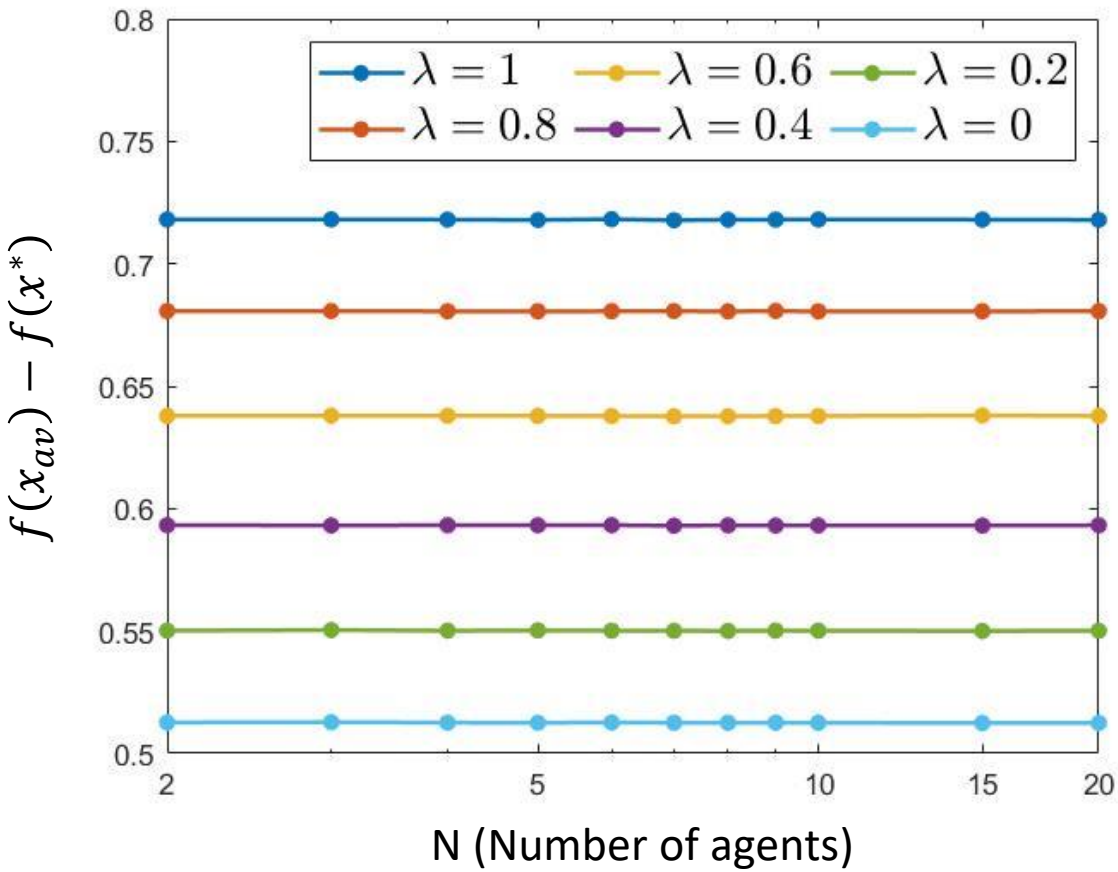
DGD Algorithm $x_i^{k+1} = \sum_{j=1}^{N} w_{ij}\, x_j^k - \alpha \nabla f_i(x_i^k)$

Settings K steps of DGD with

- Constant step-size: $\alpha = \frac{1}{\sqrt{K}}$
- Convex local functions f_i with bounded subgradients
- Identical starting points s.t. $\|x^0 - x^*\|^2 \leq 1$
- Symmetric doubly-stochastic network matrix W s.t. $\lambda(W) \in [-\lambda, \lambda]$ (except for $\lambda_1(W) = 1$)

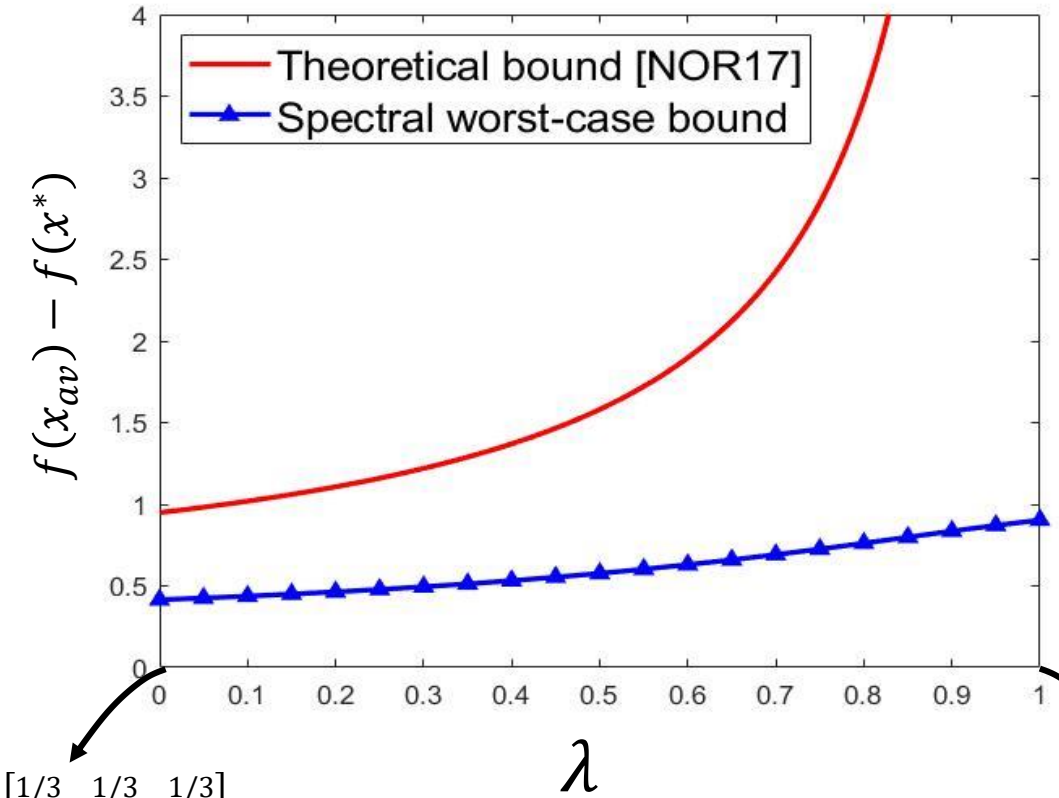
Performance criterion: $f(x_{av}) - f(x^*)$ where $x_{av} = \frac{1}{K}\sum_k \frac{1}{N}\sum_i x_i^k$

DGD – Spectral worst-case evolution with N



For $K = 5$ iterations and $\lambda(W) \in [-\lambda, \lambda]$

DGD – Spectral worst-case vs Theoretical bound



$$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

complete

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

disconnected

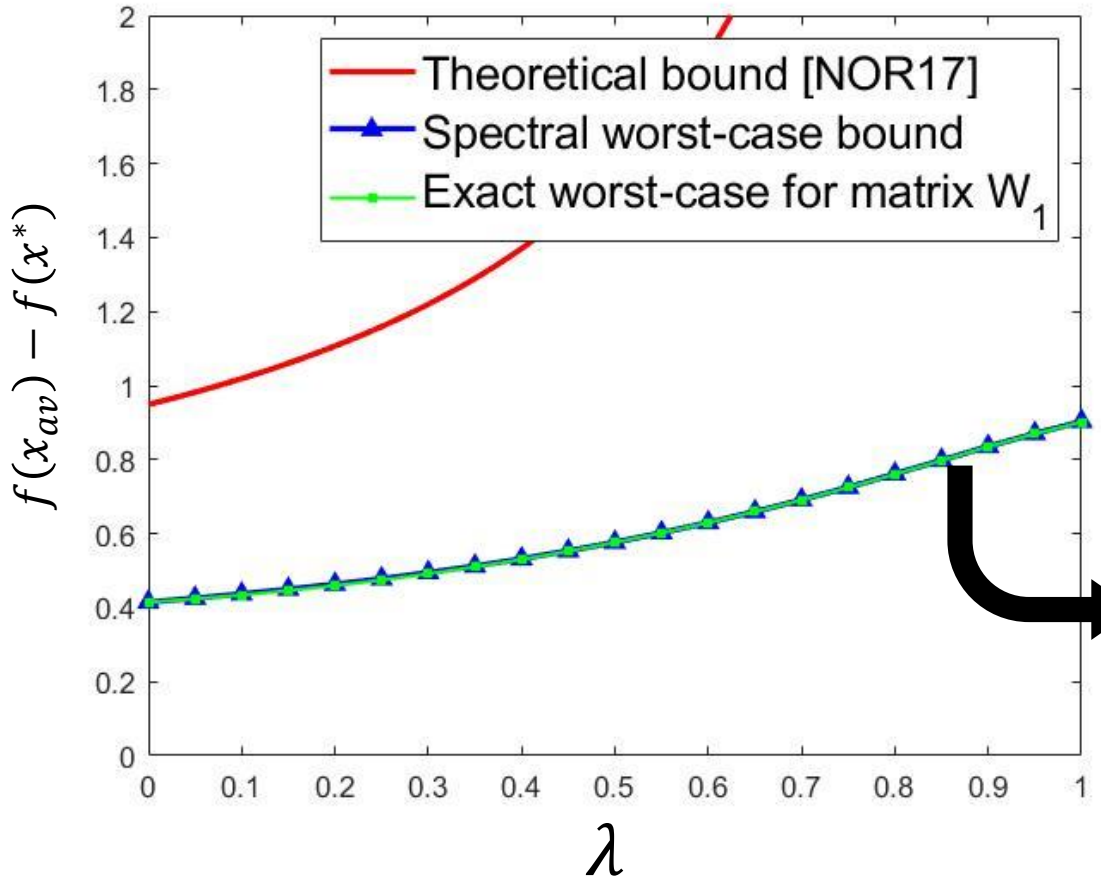
[NOR17] A. Nedic, A. Olshevsky, and M. G. Rabbat, "Network topology and communication computation tradeoffs in decentralized optimization", 2017.

Symmetric range of eigenvalues

$$-\lambda \leq \lambda_n(W) \leq \cdots \leq \lambda_2(W) \leq \lambda$$

For $K = 10$ iterations, $N = 3$ agents and $\lambda(W) \in [-\lambda, \lambda]$

Tightness Analysis



Worst-case matrix estimation

$$W_1 = YX^+$$

Observation

$$W_1 = (1+\lambda)\frac{\mathbf{1}\mathbf{1}^T}{N} - \lambda I$$

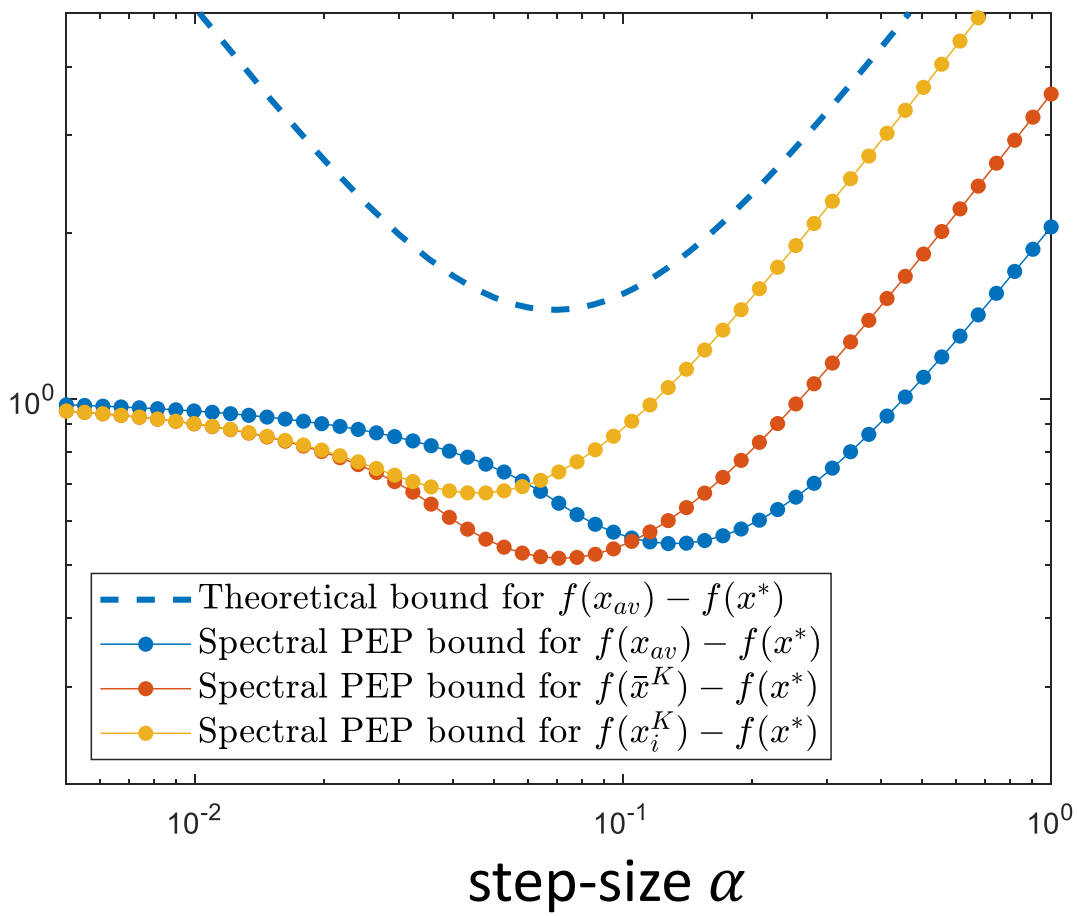
➡ $\lambda(W_1) = \{1, -\lambda, -\lambda\}$

EXACT for *spectrally* doubly-stochastic matrices

For $K = 10$ iterations, $N = 3$ agents and $\lambda(W) \in [-\lambda, \lambda]$

DGD – Tuning α

Many different performance criteria are possible

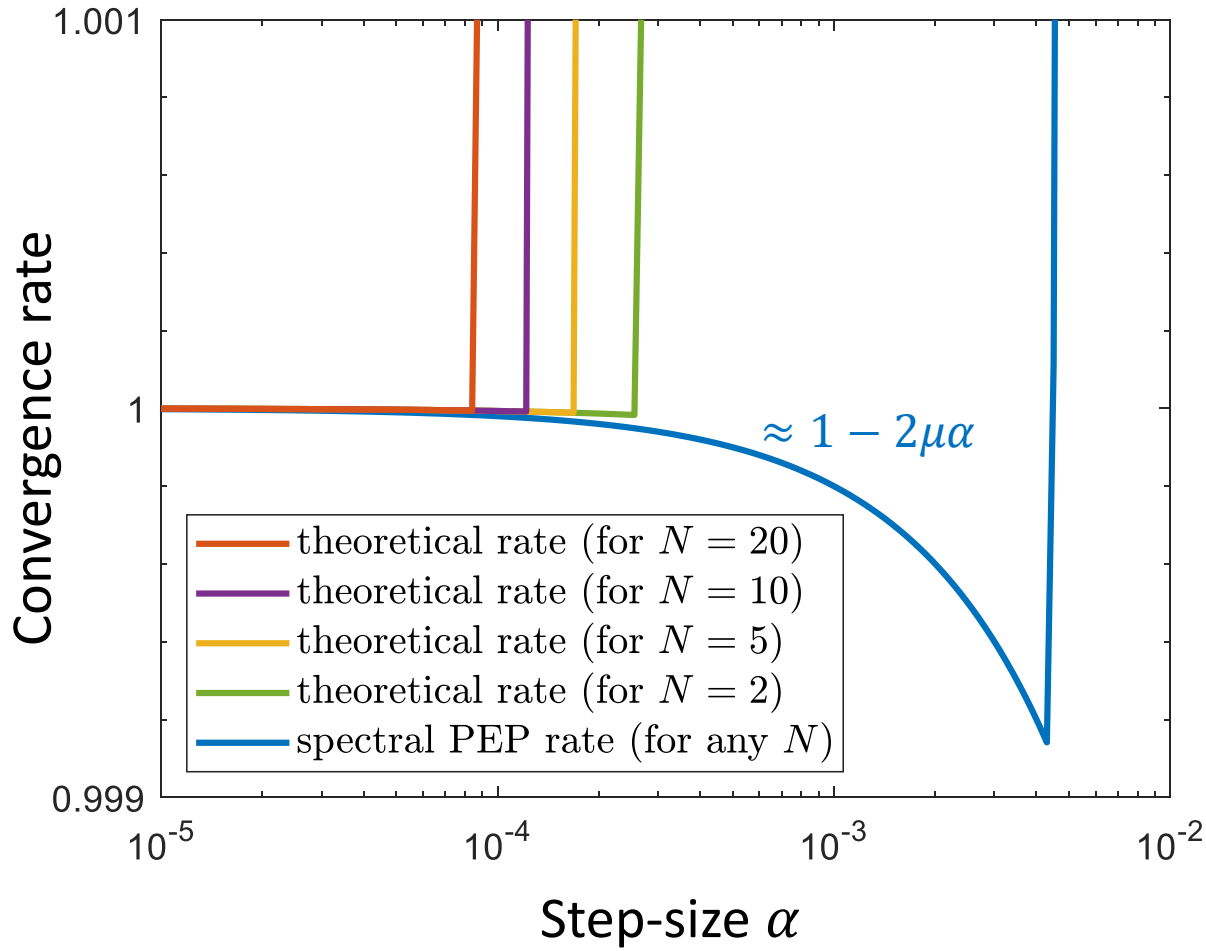


$$\bar{x}^k = \frac{1}{N}\sum_{i=1}^{N} x_i^k$$

$$x_{av} = \frac{1}{K}\sum_{k=1}^{K} \bar{x}^k$$

For $K = 10$ iterations, $N = 3$ agents and $\lambda(W) \in [-0.8, 0.8]$

Results of PEP for DIGing



For $\mu = 0.1,\ L = 1$ and $\lambda(W) \in [-0.9, 0.9]$

Computed for $N = 2$.

[NOS16] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," SIAM Journal on Optimization, 2016.

Conclusion

Sébastien Colla

Toolbox PESTO

Numerical tool for **automatic performance computation** of decentralized optimization methods

PEP idea: worst-cases are solutions of optimization problems

SPECTRAL formulation	EXACT formulation
Spectral class of matrices	Given network matrix W
Relaxation of PEP	ALWAYS exact

For DGD and DIGing:
- ✓ Independent of N
- ✓ Tight when negative weights are allowed
- ✓ Improve on the literature bounds

We can answer a large diversity of (new) questions

Future works

- ❑ Other class of networks (any suggestion?)
- ❑ Proof for the DIGing convergence rate
- ❑ Agent-independent PEP formulation

References

[CH22-A] S. Colla, J. M. Hendrickx, “Automatic Performance Estimation for Decentralized Optimization”, preprint 2022.

[CH22-B] S. Colla, J. M. Hendrickx, “Automated Performance Estimation for Decentralized Optimization via Network Size Independent Problems”, submitted to CDC 2022.

[Taylor et al.] A. B. Taylor, J. M. Hendrickx, F. Glineur, “Exact worst-case performance of first-order methods for composite convex optimization,” SIAM Journal on Optimization, 2015

[NOR17] A. Nedic, A. Olshevsky, and M. G. Rabbat, “Network topology and communication computation tradeoffs in decentralized optimization”, 2017.

[NOS16] A. Nedic, A. Olshevsky, and W. Shi, “Achieving geometric convergence for distributed optimization over time-varying graphs,” SIAM Journal on Optimization, 2016.