Consensus and distributed estimation

Sandro Zampieri
Universita' di Padova

Outline

- Problem description and scientific context
- Multi-agent systems: a distributed estimation and control architecture
- Description of the linear consensus algorithm
- Examples of applications
- Performance analysis
- Time varying consensus algorithm and its performance analysis
- Conclusions

Networked control systems

Wireless sensor networks

Swarm robotics

Communication networks

Smart grids

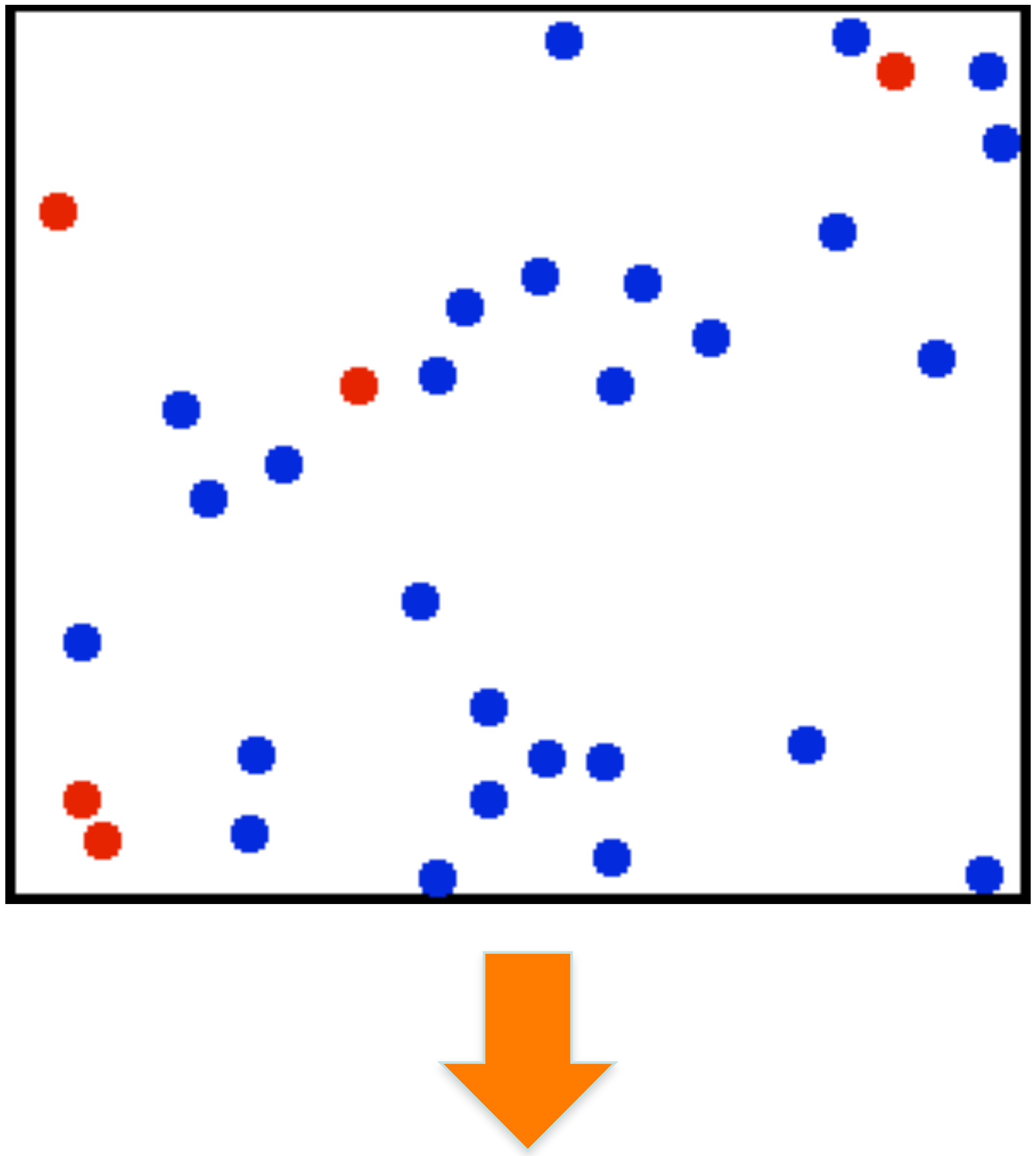
Water distribution

Traffic control

Scientific context

Statistical mechanics

how the local interactions of particles may yield simple thermodynamics laws describing the global behavior.



$$pV = nRT$$

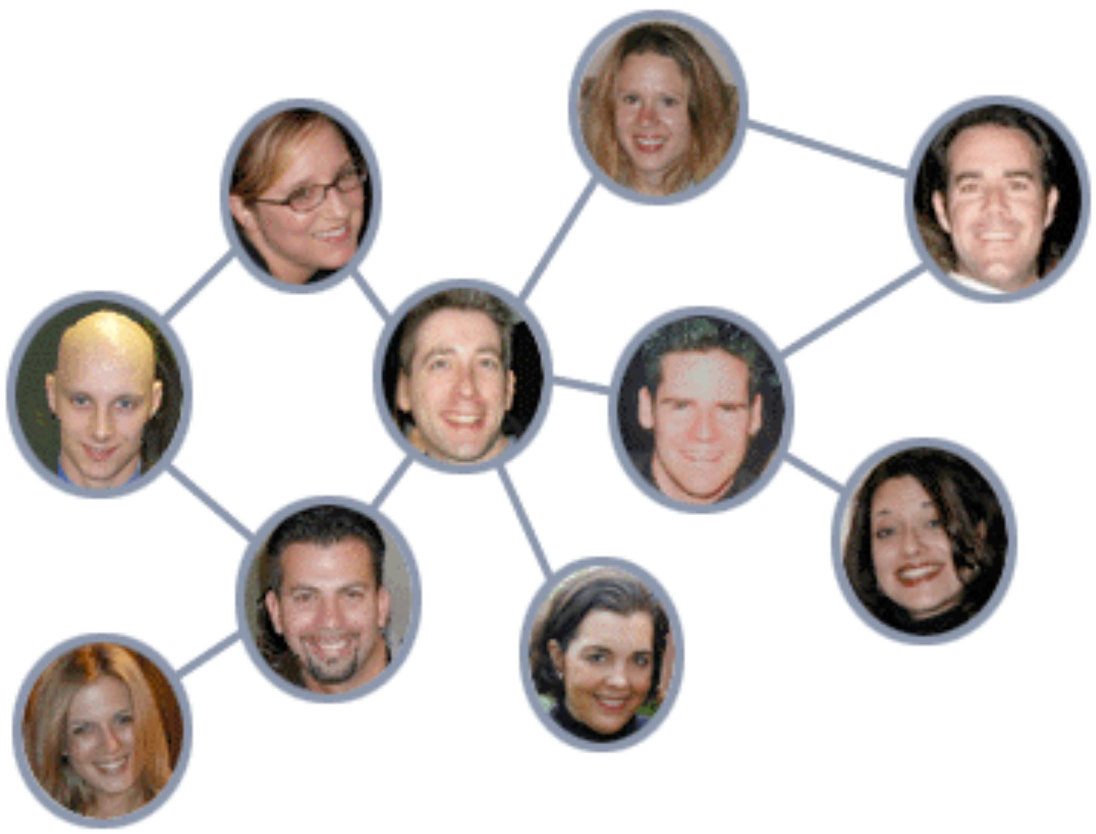
Scientific context

COOPERATION: Simple global behavior from local interactions

Flocking: collective animal behavior given by the motion of a large number of coordinated individuals

Scientific context

Social and economic networks: individual social and economic interactions produce global phenomena

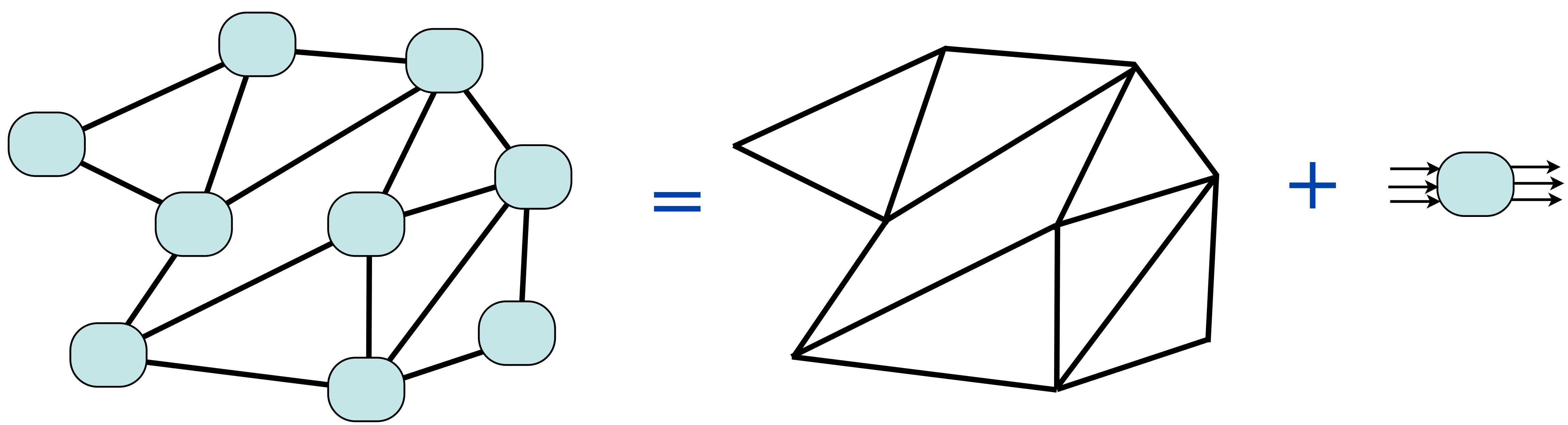


Graph describing friendship relations in an high school

Problem description

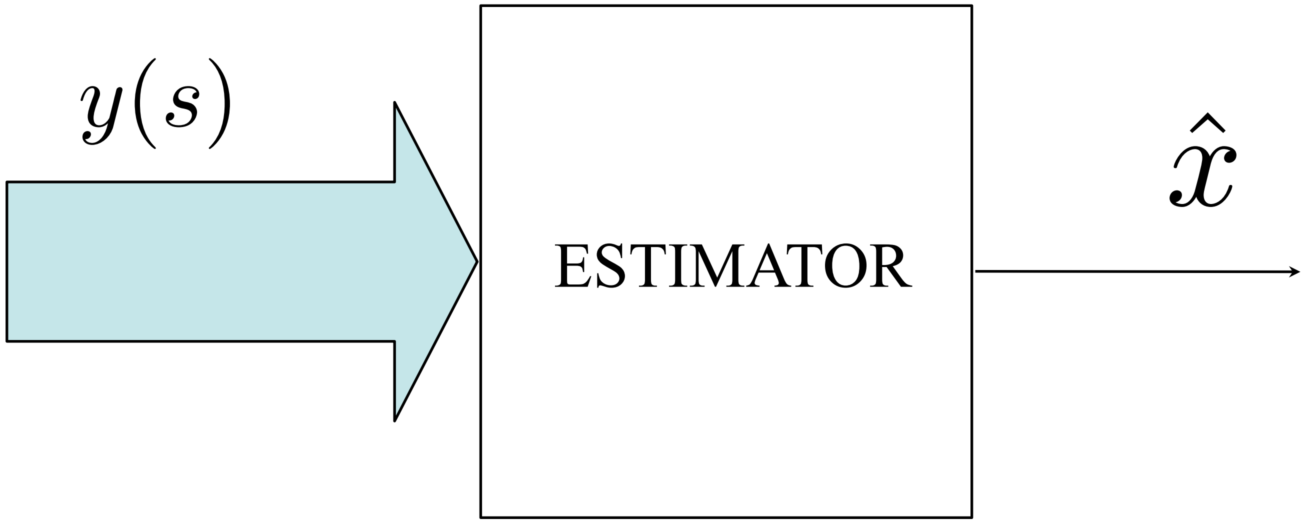
The object of our investigations is to study the behavior of "complex" systems constituted by the interconnection of many units which are themselves dynamical systems.

The behavior of these systems will depend on the dynamics of the units and on the interconnection topology. We want to understand how these two features produce the global dynamics.



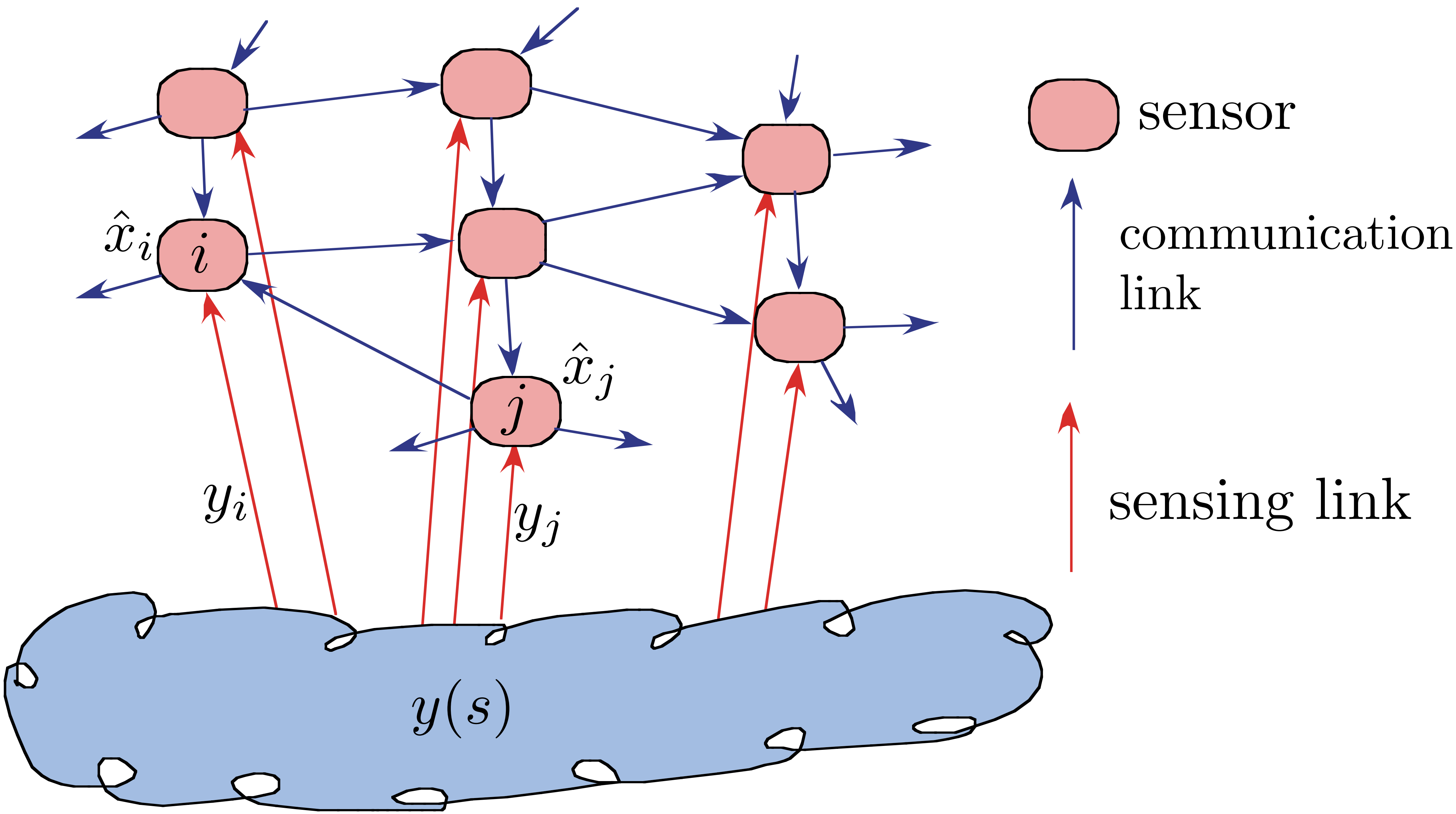
A remarkable solution of a question of this type (stability) can be found in
JA Fax, RM Murray - IEEE Transactions on Automatic Control, 2004 [Information flow and cooperative control of vehicle formations]

Multi-agent systems architecture for distributed estimation



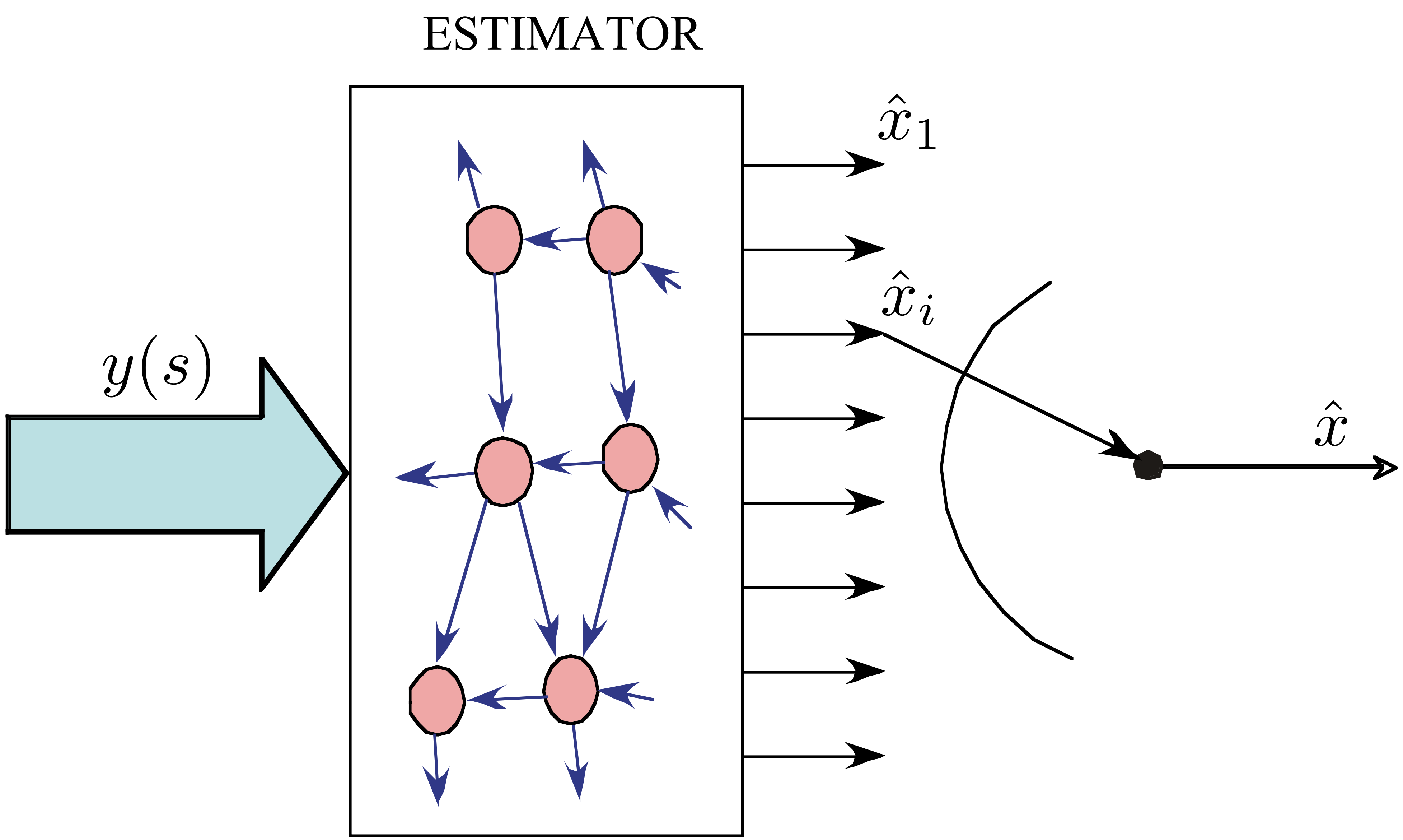
s = space variable
$y(s)$ = spatial data
$\hat{x}$ = data based decision

Multi-agent systems architecture for distributed estimation



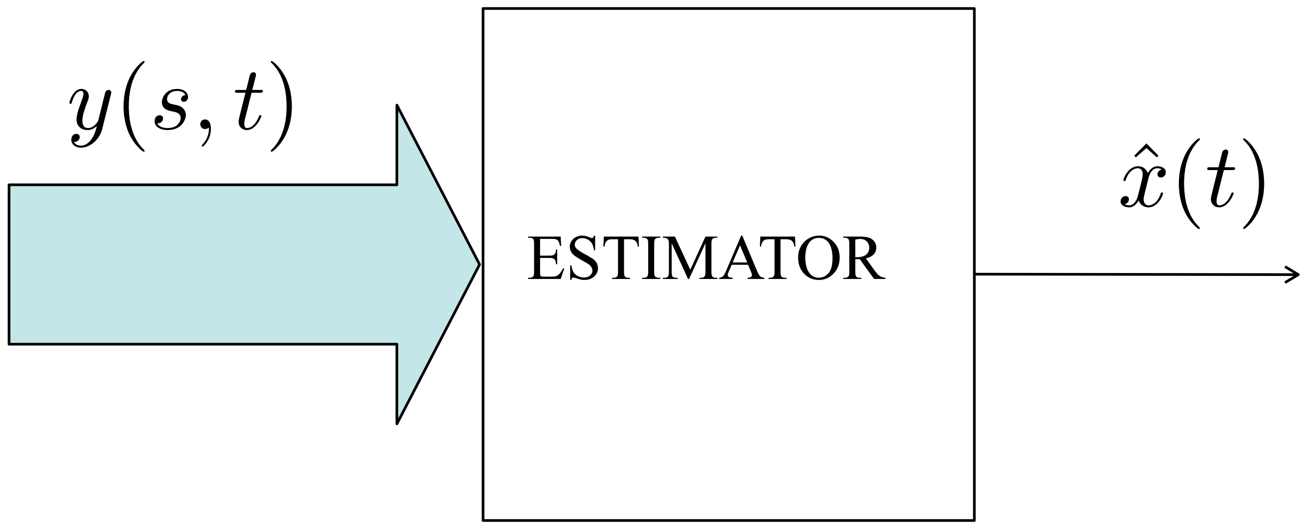
$\hat{x}_i$ is opinion of the node i has of $\hat{x}$

Multi-agent systems architecture for distributed estimation



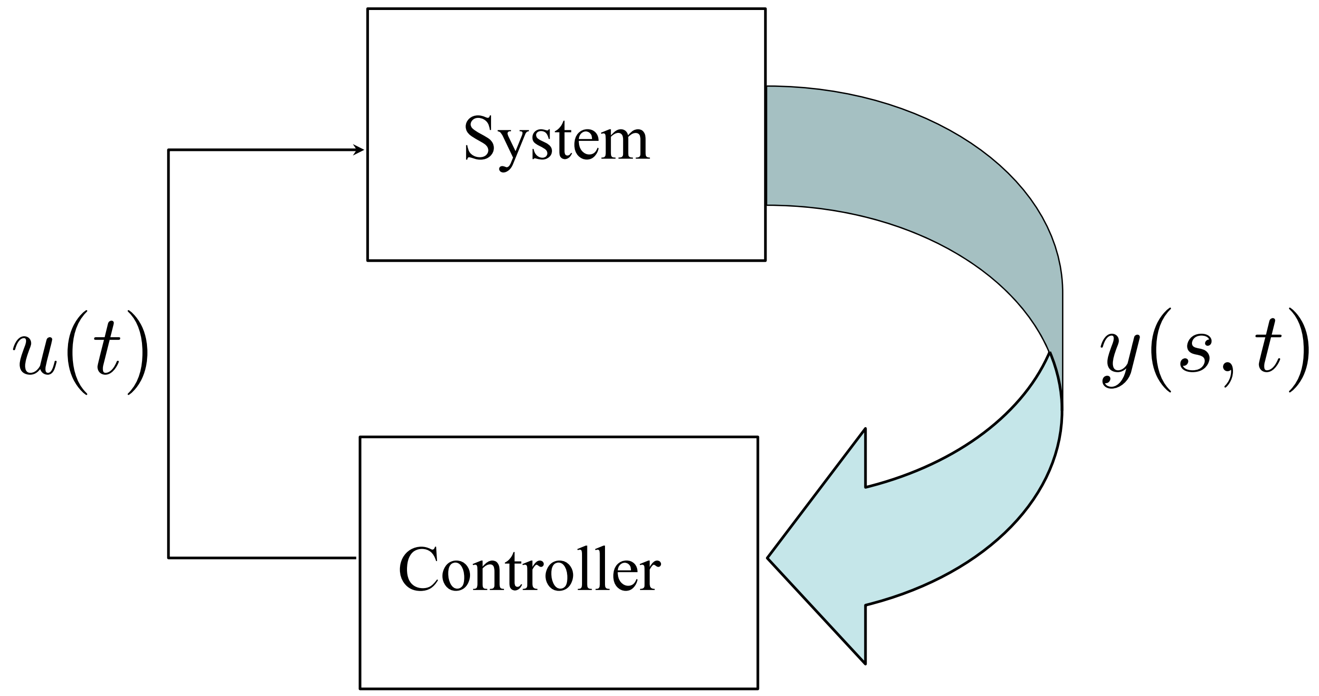
Advantages: intrinsic robustness and adaptivity due to redundancy

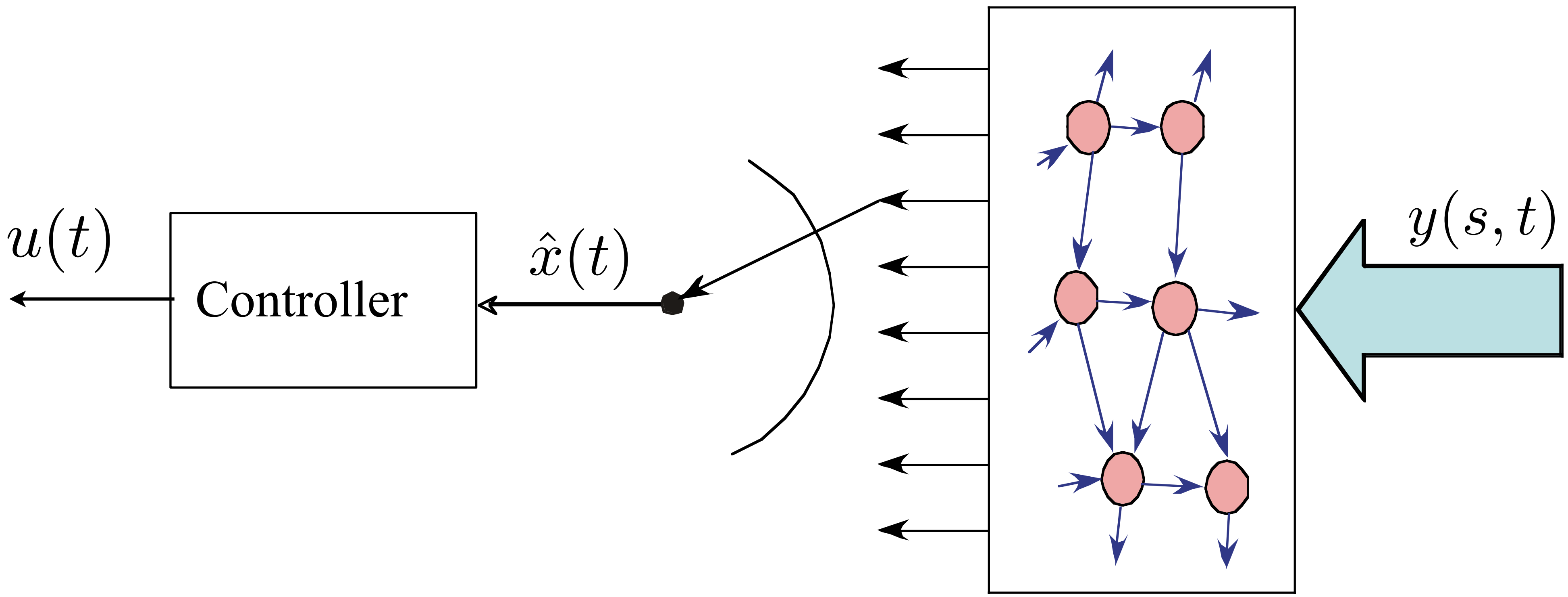
Multi-agent systems architecture for distributed estimation



t = time
s = space variable
$y(s,t)$ = time-varying spatial data
$\hat{x}(t)$ = time-varying data base decision

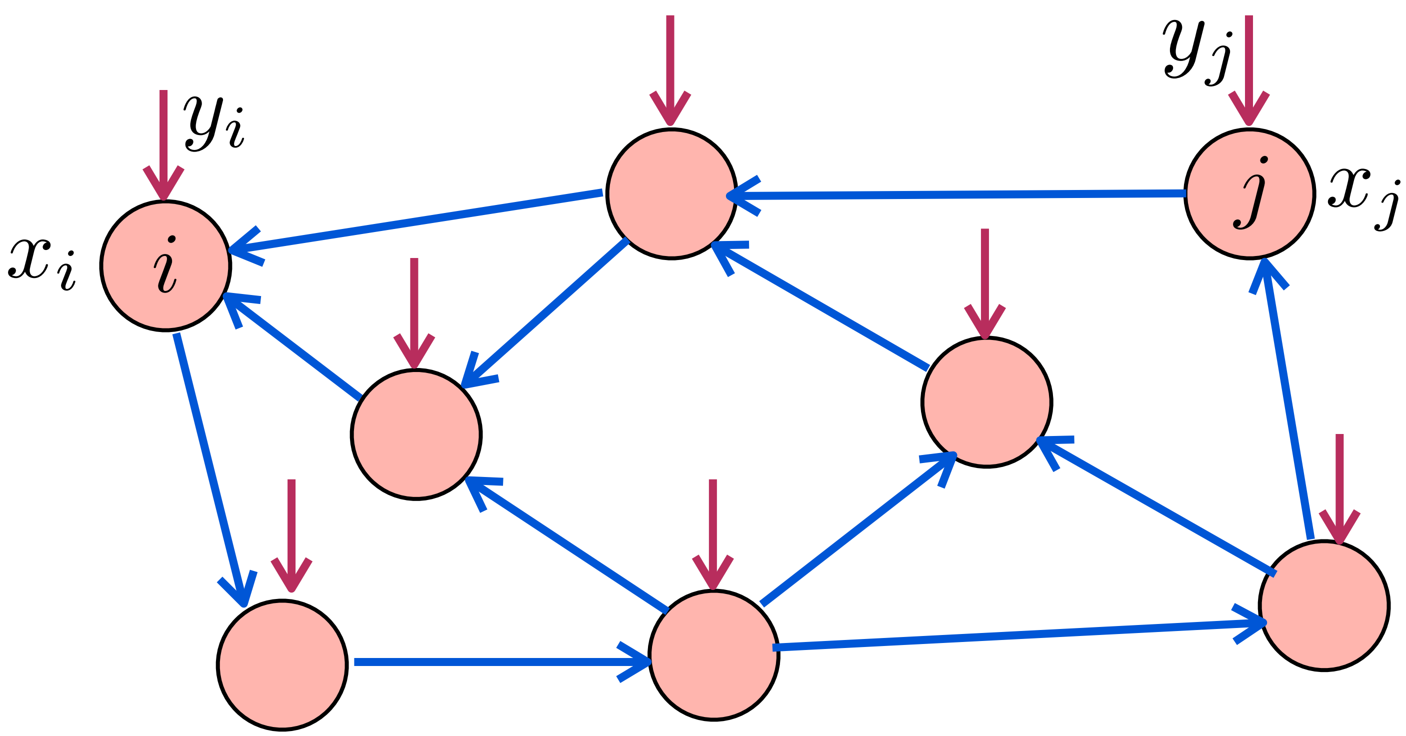
Multi-agent systems architecture for distributed estimation





The consensus algorithm

Main idea: Having a set of agents to agree upon a certain value (usually global function) using only local information exchange (local interaction)



Distributed computation of general functions

- Computational efficient (linear & asynchronous)
- Independent of graph topology
- Incremental (i.e. anytime)
- Robust to failure

x_i is the estimates of x of the node i

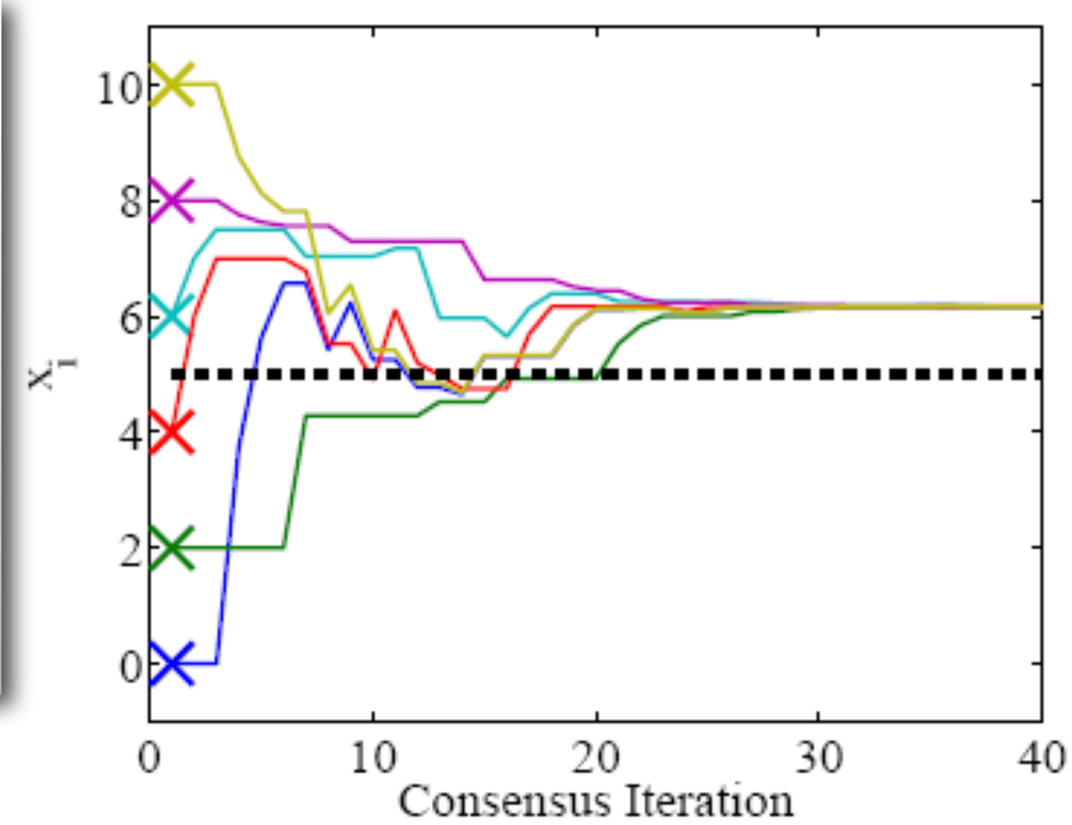
$$x = f(y_1, \ldots, y_N) = F\left(\frac{1}{N}\sum_{i=1}^{N} G_i(y_i)\right)$$

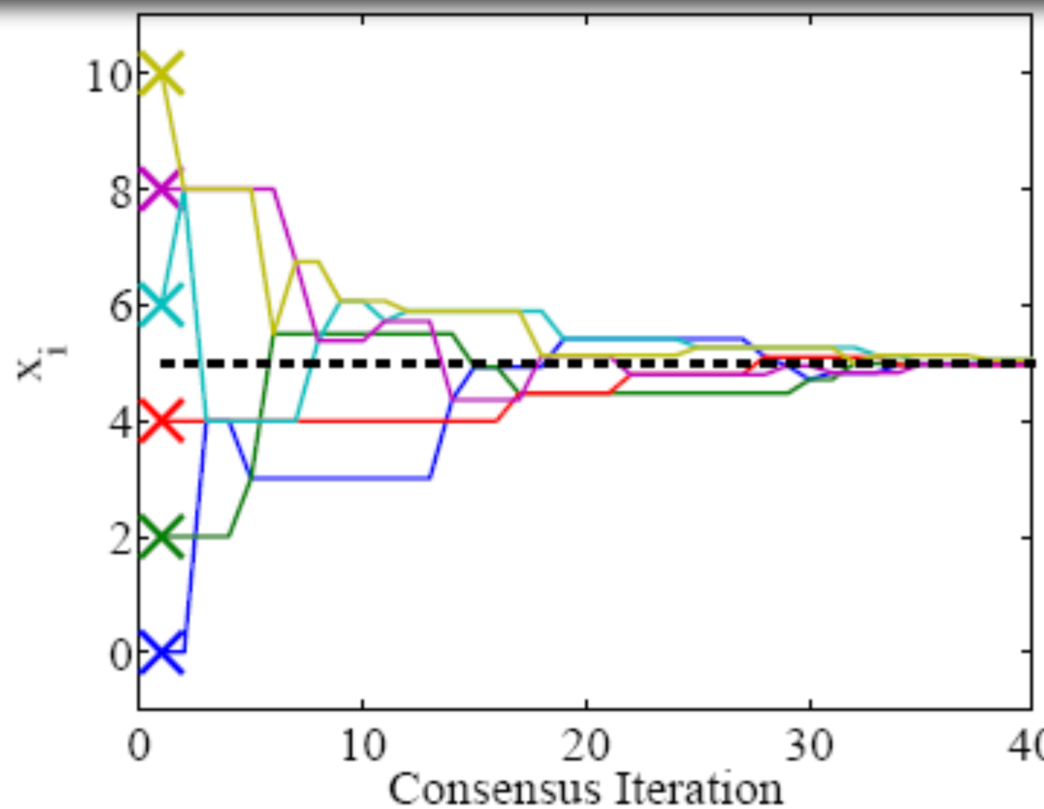
The consensus algorithm

A distributed algorithm is said to reach the consensus if

$$x_i(t) \longrightarrow \alpha$$

for all $i = 1, \ldots, N$.





A distributed algorithm is said to reach the average consensus if

$$x_i(t) \longrightarrow \frac{1}{N}\sum_{i=1}^{N} y_i$$

for all $i = 1, \ldots, N$.

The consensus algorithm

GOAL: each node has to obtain the average of $y_1, \ldots, y_N$, where y_i is known only by the node i. This task has to be performed in a distributed way.

ALGORITHM: Each sensor produces at time t an estimate $x_i(t)$ of the average as follows

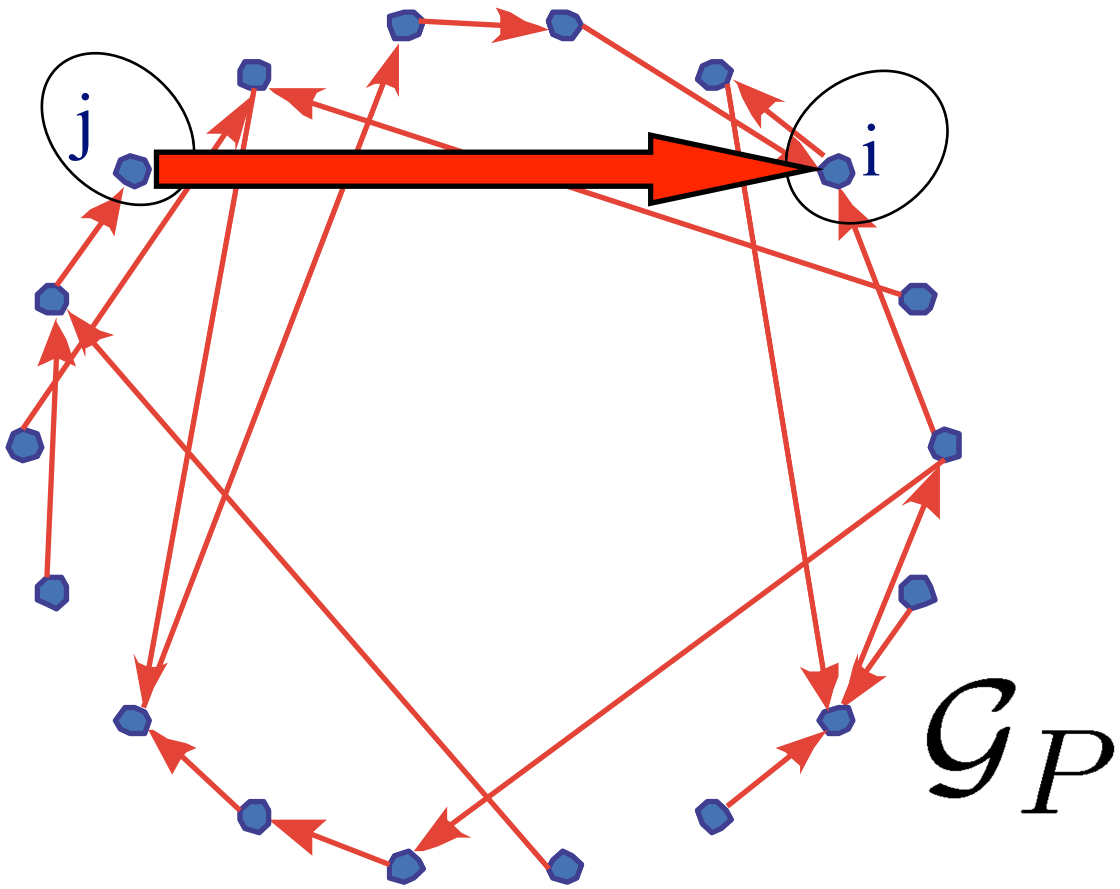
$$\boxed{x_i(t+1) = \sum_{j=1}^{N} P_{ij} x_j(t) \qquad x_i(0) = y_i}$$

COMMUNICATION: $x_j(t)$ needs to be transmitted from the node i to the node j iff

$$P_{ij} \neq 0$$

The consensus algorithm

$$x(t+1) = Px(t)$$
$$x(0) = y$$



Assume P stochastic. If $P_{ii} > 0$ for all i and the graph $\mathcal{G}_P$ associated with P is strongly connected, then all estimates converge to the same value (consensus)

$$x_i(t) \longrightarrow \sum_{j=1}^{N} \mu_j x_j(0)$$

where the weights μ_j are nonnegative and sum to one.

The consensus algorithm

In case P^T is stochastic as well (for instance if P is symmetric), then $\mu_j = 1/N$ and so

$$x_i(t) \longrightarrow \frac{1}{N}\sum_{j=1}^{N} \mu_j x_j(0)$$

This idea can be used as a way for computing averages and more in general to compute functions like

$$\boxed{f(y_1, \ldots, y_N) = F\left(\frac{1}{N}\sum_{i=1}^{N} G_i(y_i)\right)}$$

- Computational efficient (linear & asynchronous)
- Independent of graph topology
- Incremental (i.e. anytime)
- Robust to failure

Some literature (limited to the control field)

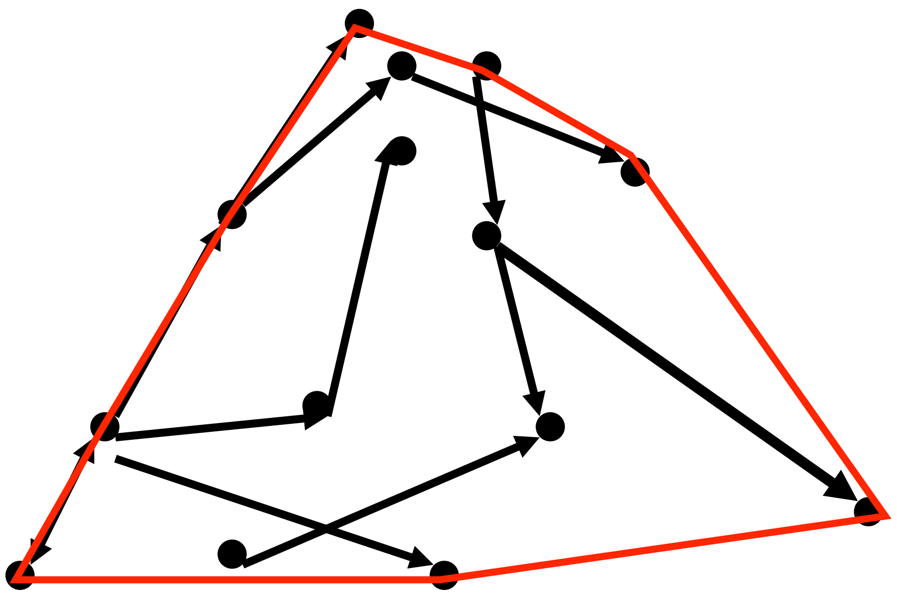
- **Convergence of Markov Chains (60's) and Parallel Computation Alg.(70's)**
- John Tsitsiklis "Problems in Decentralized Decision Making and Computation ", Ph.D thesis, 1984
- Time-varying topologies (deterministic worst-case analysis)
  - **Convergence**: Moreau (2005), Jadbabaie, Lin, Morse (2002), Olfati Saber, Murray (2004), Cao, Morse, Anderson (2008), ...
- Randomized consensus
  - **Convergence**: Y. Hatano and M. Mesbahi,(2005), Wu (2006), Boyd, Ghosh, Prabhakar, Sha (2006), Alireza Tahbaz-Salehi, Ali Jadbabaie (2006),
  - **Performance**: Boyd, Ghosh, Prabhakar, Sha (2006), Patterson, Bamieh, Abbadi (2007), Fagnani, Zampieri, (2007)
- Applications
  - **Vehicle coordination:** many contributions
  - **Distributed Kalman Filtering**: Xiao, S. Boyd, and S. Lall. (2005), Olfati Saber (2005), Alighanbari, How (2006), Carli, Chiuso, Schenato, Zampieri (2008), Alriksson, Rantzer (2006), Spanos, R. Olfati-Saber, and R. M. Murray.(2005), I.D. Schizas, G.B. Giannakis, S. I. Roumeliotis, and A Ribeiro.(2007), A. Speranzon, C. Fischione, and K. Johansson (2006)
  - **Generalized means**: Bauso, L. Giarre', and R. Pesenti (2006),Cortes (2008)
  - **Time-synchronization**: Solis, Borkar, Kumar (2006), Simeone, Spagnolini (2007), Carli, Chiuso, Schenato, Zampieri (2008), Schenato, Fiorentin (2009)
  - **Sensor and camera network calibration**: Barooah, Hespanha (2005) ,Bolognani, DelFavero, Schenato, Varagnolo (2008), Tron, Vidal (2009)

Some literature on performance metrics

- Rate of convergence, mixing rate of a Markov chain, spectral gap and essential spectral radius of a stochastic matrix (from the 70's)
  - **Cayley graphs**: Diaconis (1990-2000), Carli, Fagnani, Speranzon, Zampieri (2008).
  - **Random geometric graph**: Boyd, Ghosh, Prabhakar, Sha (2006).
  - **Performance** Classical literature of Markov chains (Diaconis, Stroock), Xiao, Boyd (2006), B. Bamieh, M. Jovanovic, P. Mitra, and S. Patterson. (2010)

- L2 performance metrics
  - **General considerations**: Xiao, Boyd, Lall, Diacomis, Kim (2004-2009).
  - **Cayley graphs**: Bamieh, Javonovic, Mitra, Patterson (2009), Carli, Z. (2008), Garin, Zampieri (2009).
  - **Random geometric graph**: Barooah, Hespanha (2004-2009), Carli, Lovisari, Zampieri (2010).

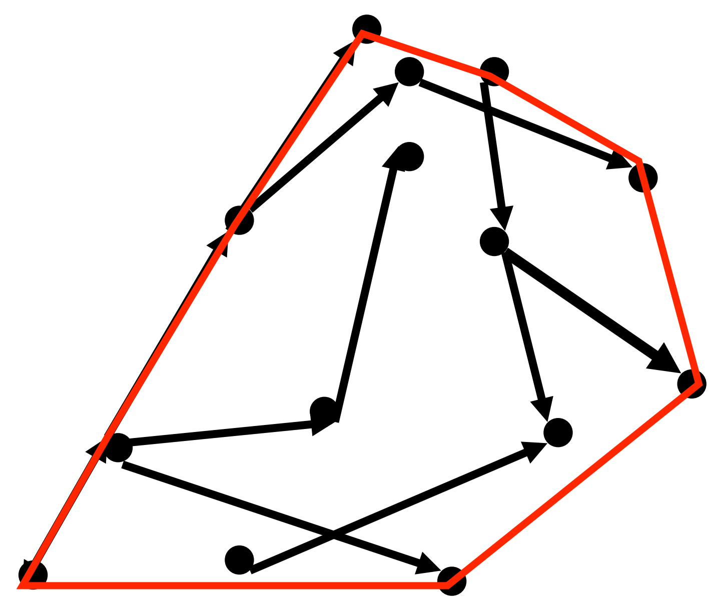
The consensus algorithm



Idea for the proof of convergence:

Convex hull always shrinks.
If communication graph sufficiently connected, then shrinks to a point

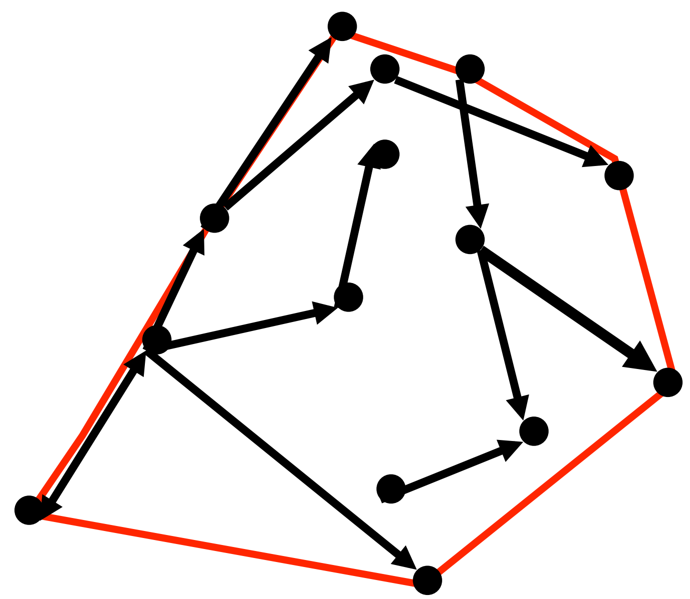
The consensus algorithm



Idea for the proof of convergence:

Convex hull always shrinks.
If communication graph sufficiently connected, then shrinks to a point

The consensus algorithm



Idea for the proof of convergence:

Convex hull always shrinks.
If communication graph sufficiently connected, then shrinks to a point

Variations of the consensus algorithm

- $\boxed{\text{Continuous time}}$

$$\dot{x}(t) = Kx(t)$$

where K is a Laplacian of a graph (the matrix $I - \epsilon K$ is stochastic).

- $\boxed{\text{Time-varying case}}$

$$x(t+1) = P(t)x(t)$$

- $\boxed{\text{Randomly time-varying case}}$

$$x(t+1) = P(t)x(t)$$

in which $P(t)$ is a matrix values stochastic process, e.g. independent and identically distributed.

Variations of the consensus algorithm

$\boxed{\text{Higher order}}$ (Synchronization) $x_i(t) \in \mathbb{R}^n$

$$x_i(t+1) = Ax_i(t) + Bu_i(t)$$
$$y_i(t) = Cx_i(t)$$

with feedback control

$$u_i(t) = \sum_{j=1}^{N} K_{ij} y_j(t)$$

where $K \in \mathbb{R}^{N \times N}$ is the Laplacian of a graph.

$$y_i(t) \longrightarrow \alpha$$

Variations of the consensus algorithm

$\boxed{\text{Nonlinear}}$ (Synchronization) $x_i(t) \in \mathbb{R}^n$

$$\begin{aligned} x_i(t+1) &= f(x_i(t)) + g(x_i(t))u_i(t) \\ y_i(t) &= h(x_i(t)) \end{aligned}$$

with feedback control

$$u_i(t) = \sum_{j=1}^{N} K_{ij} y_j(t)$$

where $K \in \mathbb{R}^{N \times N}$ is the Laplacian of a graph.

$$y_i(t) \longrightarrow \alpha$$

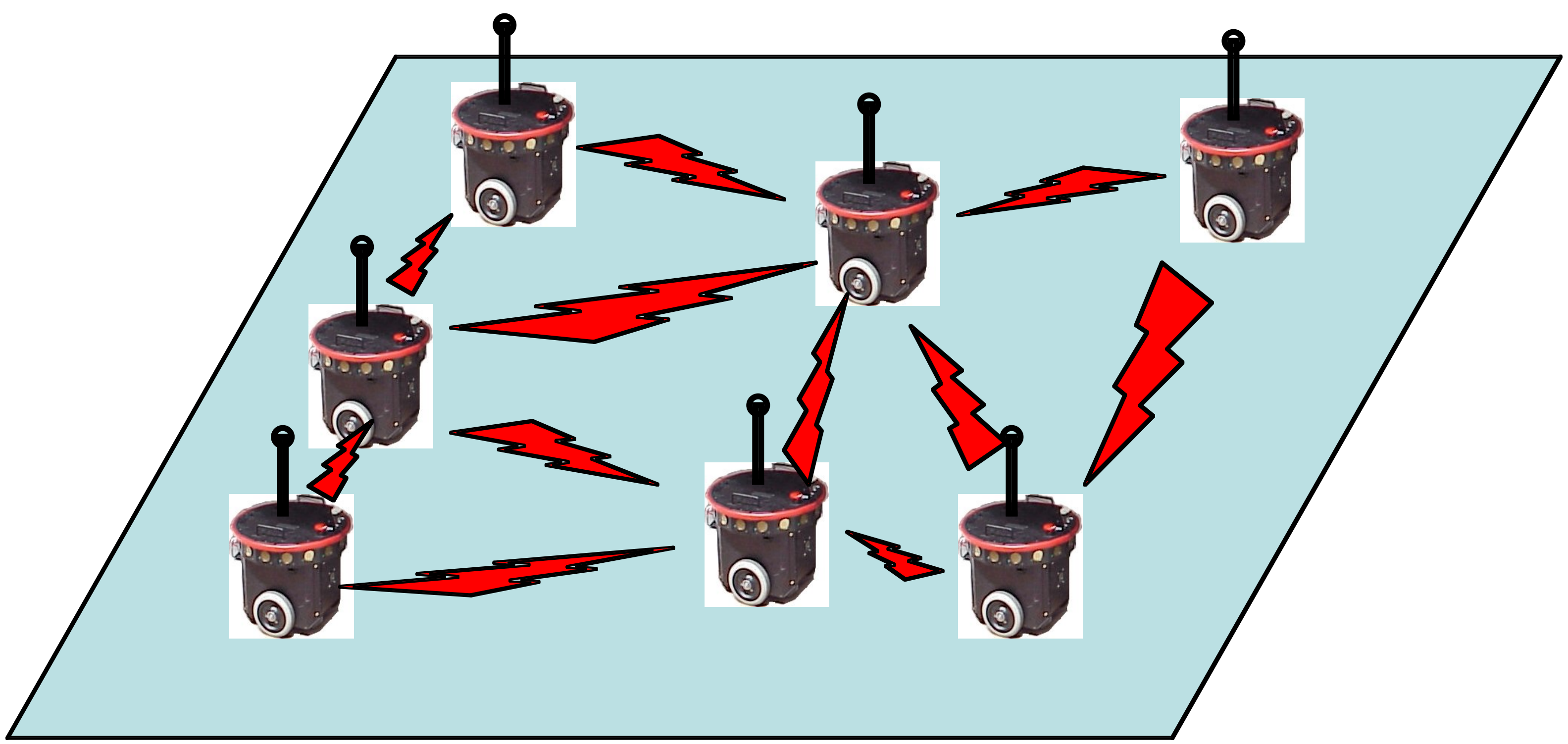
Example: vehicle formation

Assume we have N vehicles moving on the plane. Each vehicle has coordinates $z_i(t) = (x_i(t), y_i(t))^T$. The goal is the rendezvous of the vehicles in one point of the plane (can be generalized to formation reaching).

Solution:

$$\boxed{z_i(t+1) = \sum_{i=1}^{N} P_{ij} z_j(t)}$$

The vehicles will reach asymptotically the centroid on the initial positions.



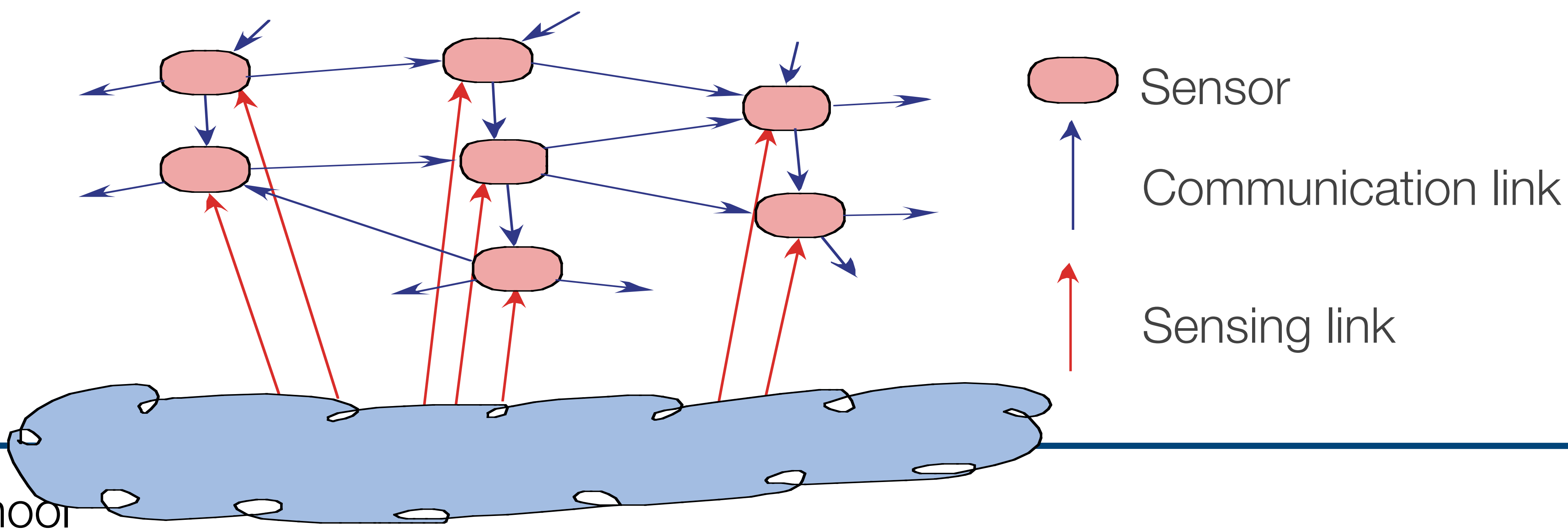
Example: distributed estimation

Assume that N sensors have to estimate a quantity $x \in \mathbb{R}$ from their noisy measurements. The result of the measurement of the sensor i is

$$\boxed{y_i = x + n_i}$$

where n_i are independent noises of zero mean and variance 1. The best estimate of x from the measurements is

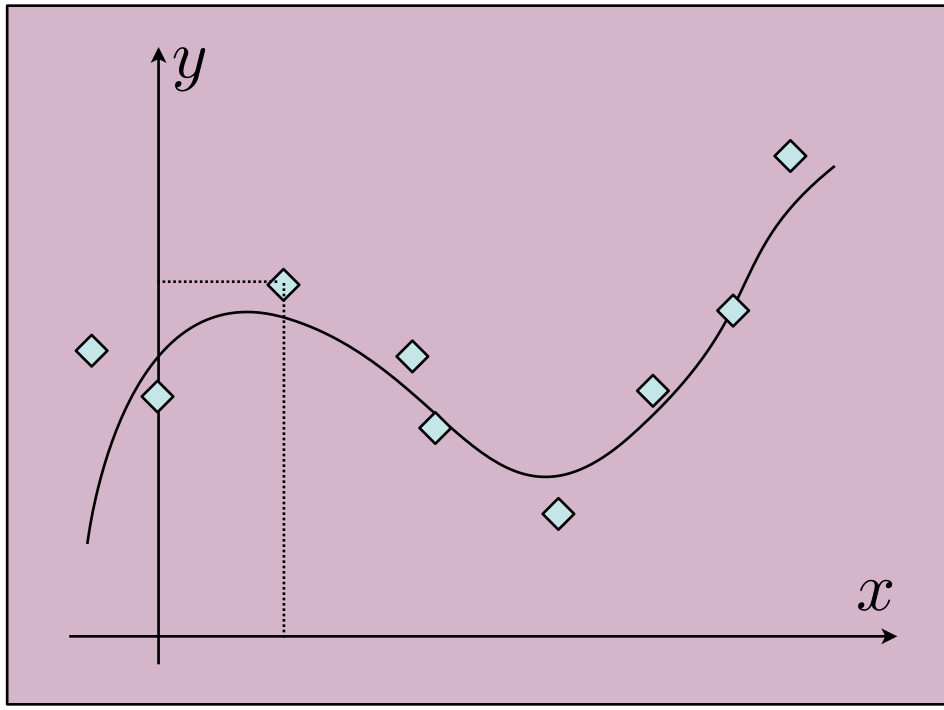
$$\boxed{\hat{x} := \frac{1}{N} \sum_i y_i}$$



Example: distributed least square

Assume that each sensor i measures two variables x_i, y_i and that the relation between these needs to be estimated. The relation is modeled by a finite dimensional function space

$$\boxed{f(x) = \sum_{i=1}^{n} \theta_i f_i(x)}$$



where the functions $f_i(x)$ form the basis of the function space. We need to estimate the coefficients θ_i. We can write

$$f(x) = F^T(x)\Theta$$

where

$$F^T(x) = [f_1(x) \ \cdots \ f_n(x)] \qquad \Theta = [\theta_1 \ \cdots \ \theta_n]^T$$

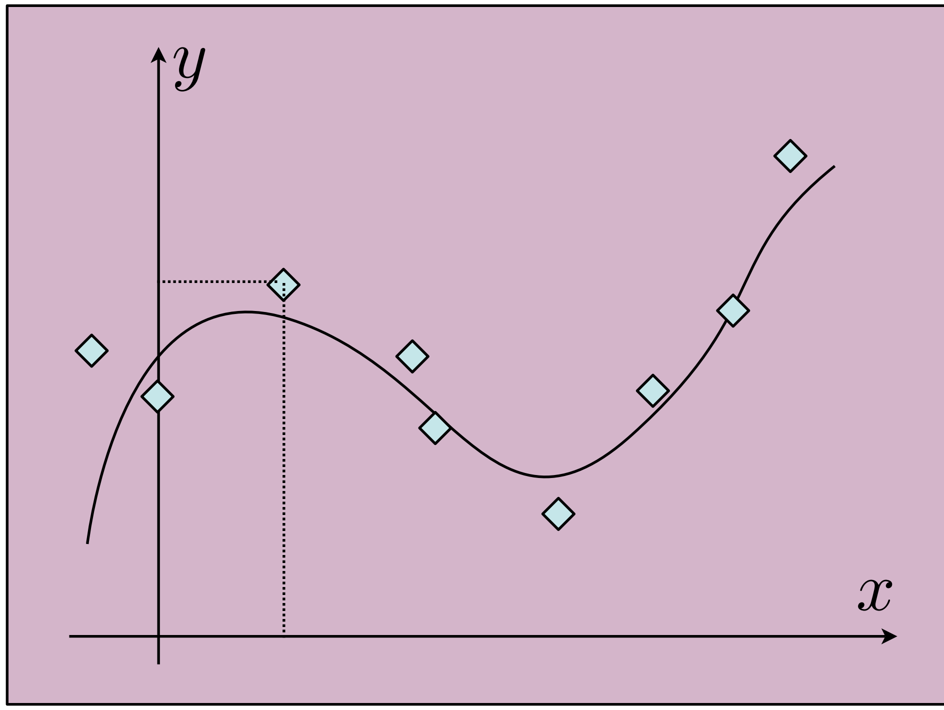
PROBLEM: Determine

$$\boxed{\hat{\Theta} := \mathrm{argmin}_{\Theta} \sum_{j=1}^{N} (y_i - F^T(x_i)\Theta)^2}$$

Example: distributed least square

According the theory of least square optimization we have that

$$\boxed{\hat{\Theta} = \left(\frac{1}{N}\sum_{i=1}^{N} F(x_i)F^T(x_i)\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} F(x_i)y_i\right)}$$



SOLUTION

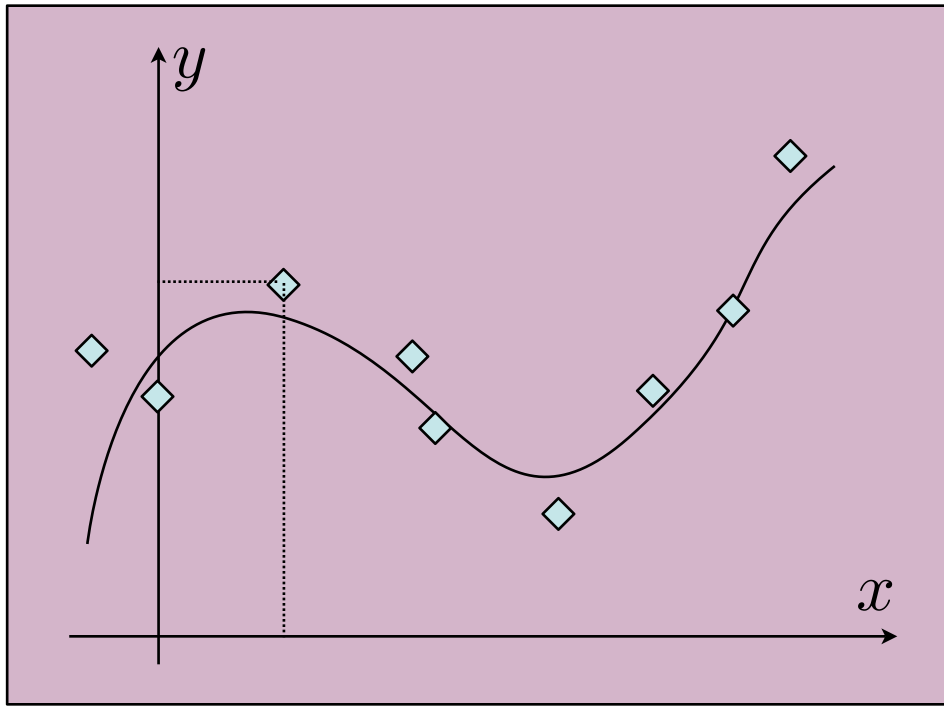
$$M_i(0) = F(x_i)F^T(x_i) \in \mathbb{R}^{n\times n} \xrightarrow{\text{average consensus}} M_i(\infty) = \frac{1}{N}\sum_{i=1}^{N} F(x_i)F^T(x_i)$$

$$v_i(0) = F(x_i)y_i \in \mathbb{R}^{n} \xrightarrow{\text{average consensus}} v_i(\infty) = \frac{1}{N}\sum_{i=1}^{N} F(x_i)y_i$$

Example: distributed least square

According the theory of least square optimization we have that

$$\boxed{\hat{\Theta} = \left(\frac{1}{N} \sum_{i=1}^{N} F(x_i) F^T(x_i) \right)^{-1} \left(\frac{1}{N} \sum_{i=1}^{N} F(x_i) y_i \right)}$$



SOLUTION

$$M_i(0) = F(x_i)F^T(x_i) \in \mathbb{R}^{n \times n} \xrightarrow{\text{average consensus}} M_i(\infty) = \frac{1}{N} \sum_{i=1}^{N} F(x_i) F^T(x_i)$$

Initial knowledge of the node i

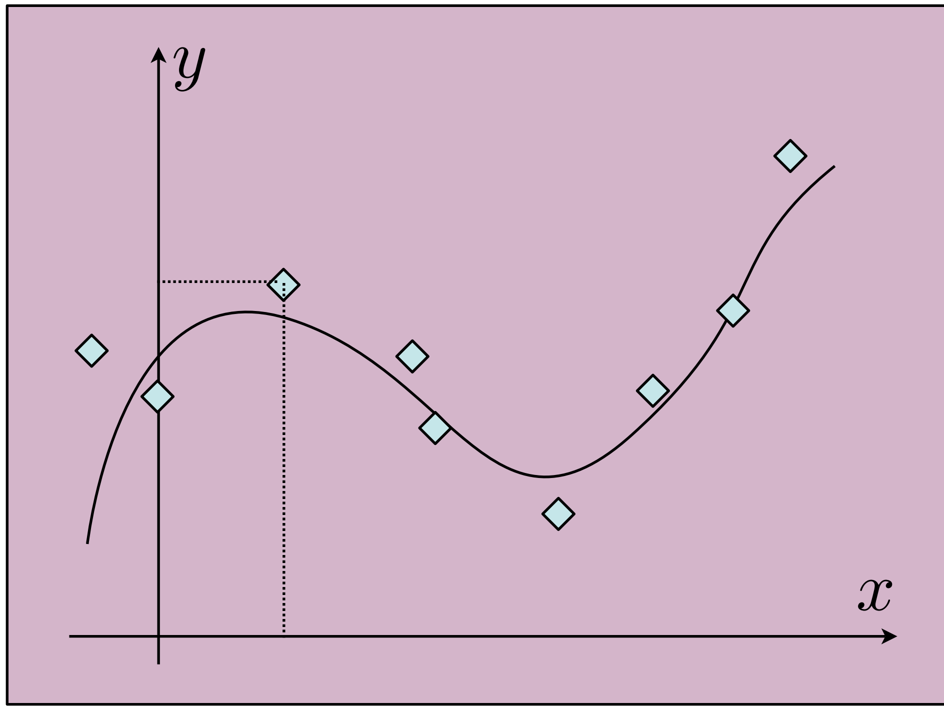
Final knowledge of the node i

$$v_i(0) = F(x_i) y_i \in \mathbb{R}^{n} \xrightarrow{\text{average consensus}} v_i(\infty) = \frac{1}{N} \sum_{i=1}^{N} F(x_i) y_i$$

Example: distributed least square

According the theory of least square optimization we have that

$$\boxed{\hat{\Theta} = \left(\frac{1}{N}\sum_{i=1}^{N} F(x_i)F^T(x_i)\right)^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} F(x_i)y_i\right)}$$



SOLUTION

$$M_i(0) = F(x_i)F^T(x_i) \in \mathbb{R}^{n\times n} \xrightarrow{\text{average consensus}} M_i(\infty) = \frac{1}{N}\sum_{i=1}^{N} F(x_i)F^T(x_i)$$

Initial knowledge of the node i

Final knowledge of the node i

$$v_i(0) = F(x_i)y_i \in \mathbb{R}^{n} \xrightarrow{\text{average consensus}} v_i(\infty) = \frac{1}{N}\sum_{i=1}^{N} F(x_i)y_i$$

$$\boxed{\hat{\Theta} = (M_i(\infty))^{-1} v_i(\infty)}$$

Example: distributed calibration

From Barooah and Hespanha

Assume we have a graph $G = (V, E)$ with vertex set $V = \{1, \ldots, N\}$ and edge set $E \subseteq V \times V$ and also a set of numbers $b_{(i,j)}$ for all $(i,j) \in E$. We want to estimate absolute values from differences, namely

$$\boxed{\min_{z_1,\ldots,z_N} \left\{ \sum_{(i,j)\in E} C_{(i,j)} \left(z_i - z_j - b_{(i,j)}\right)^2 \right\}}$$

where $C_{(i,j)}$ are weights.

In vector form this becomes

$$\min_z ||Az - b||_C^2$$

where A is the incidence matrix of the graph (a suitable $|E| \times |V|$ matrix with $0, 1, -1$) and where $C = \mathrm{diag}\{C_e : e \in E\}$.

Example: distributed calibration

To make this problem having a unique solution we add the condition

$$\min_{z_i \mid \sum z_i = 0} ||Az - b||_C^2$$

The solution is given by A distributed way to obtain the solution z is through a "consensus" type algorithm

$$\boxed{\begin{aligned} z(t+1) &= Pz(t) + \alpha A^T C b \\ z(0) &= 0 \end{aligned}}$$

where $P = I - \alpha A^T C A$ is a symmetric stochastic matrix and where α is a positive constant. If we define $x(t) := z - z(t)$, we have that

$$\begin{aligned} x(t+1) &= Px(t) \\ x(0) &= z \end{aligned}$$

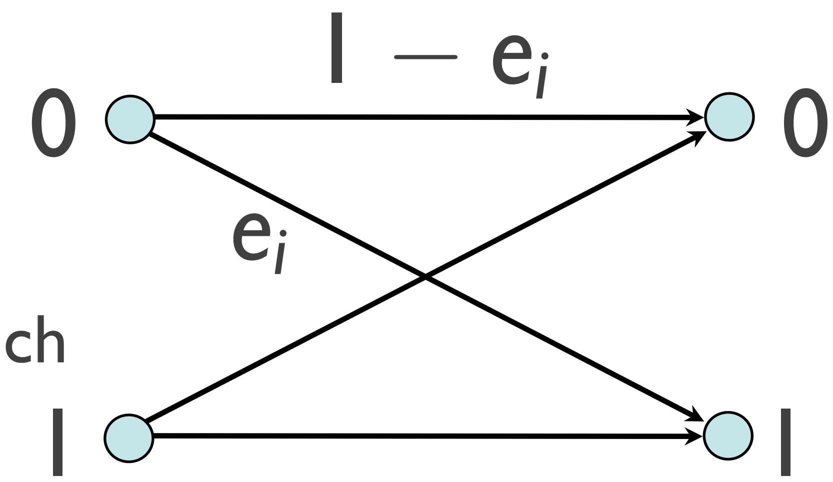
and so

$$z(\infty) = z - x(\infty) = z - \frac{1}{N}\sum_i z_i = z$$

Example: distributed decision making

We have a binary random variable x such with prior

$$P(x=0)=P(x=1)=1/2$$



N sensors can estimate x though a binary random variable y_i which are conditional independent and with conditional probabilities

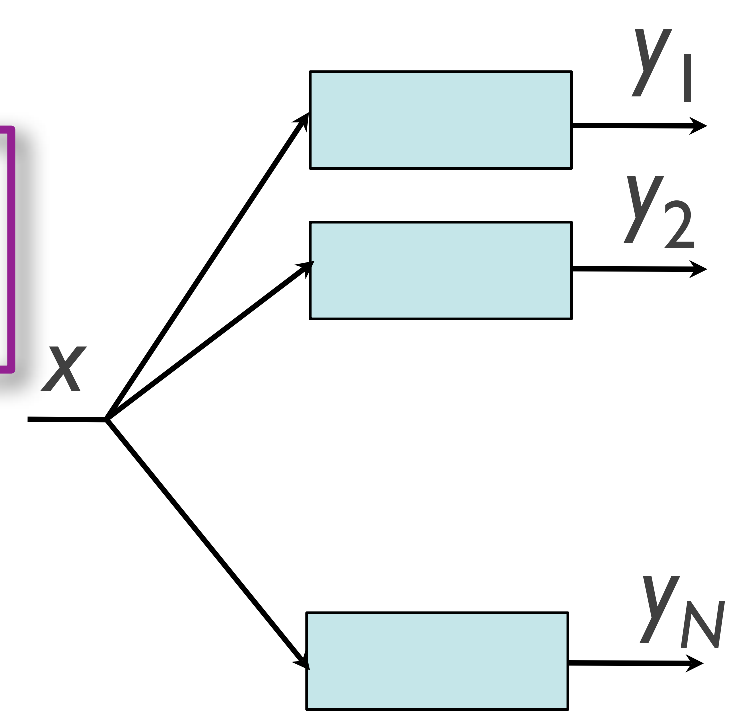
$$P(y_i=1|x=0)=P(y_i=0|x=1)=e_i$$

$$P(y_i=0|x=0)=P(y_i=1|x=1)=1-e_i$$

It can be seen that the normalized log-likelihood function is

$$\boxed{\mathcal{L}(y_1,\ldots,y_N)=\frac{1}{N}\log\frac{P(0|y_1,\cdots,y_N)}{P(1|y_1,\cdots,y_N)}=\frac{1}{N}\sum_i(1-2y_i)\log\frac{1-e_i}{e_i}}$$

$$\hat{x}=0\iff\mathcal{L}(y_1,\ldots,y_N)>0$$



Pros and cons

Advantages

1. Very robust to node and link failures.
2. Very simple implementation.
3. No need of a centralized design for the weights selection.

Disadvantages

1. Can have slow convergence for some class of communication topologies.
2. Might be sensitive to malicious nodes.

Performance metrics

$$\boxed{\begin{aligned} x(t+1) &= Px(t) \\ x(0) &= y \end{aligned}} \quad \Longrightarrow \quad \boxed{x_i(t) \longrightarrow \sum_{j=1}^{N} \mu_j x_j(0)}, \qquad \sum_{j=1}^{N} \mu_j = 1$$

Performance indices

1. Steady state performance: The difference between μ_j and $1/N$.
2. Transient performance: Speed of convergence of $x_i(t)$ to $x_i(\infty)$.

Performance metrics

$$\boxed{x_i(t) \longrightarrow \sum_{j=1}^{N} \mu_j x_j(0)}$$

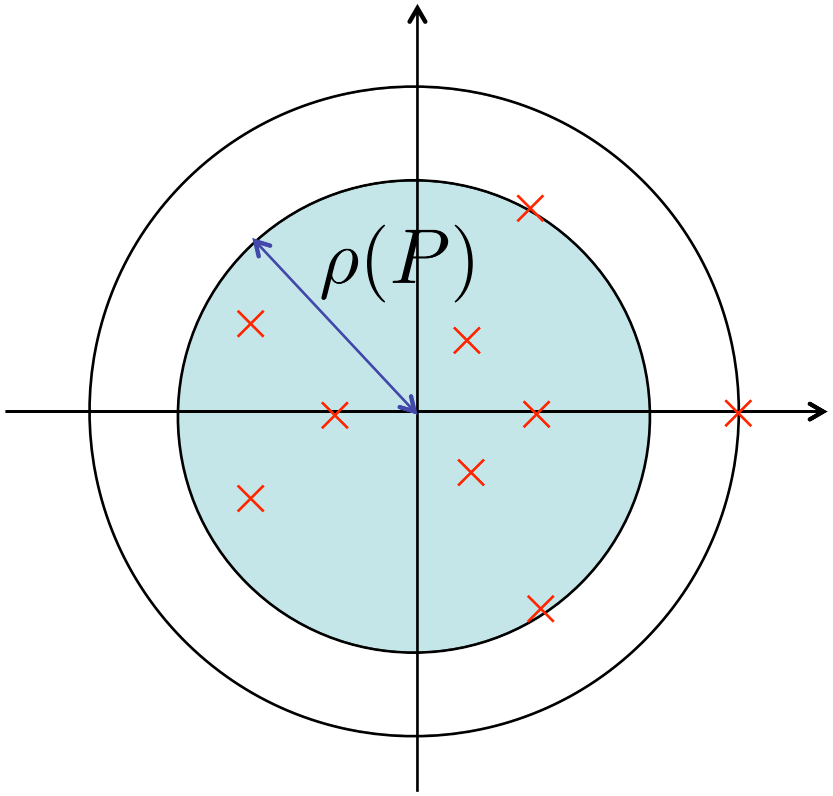
From the theory of Markov chains

1. The vector $(\mu_1, \ldots, \mu_N)$ is the invariant measure of the Markov chain. Therefore $\mu_j = 1/N$ if and only if P is doubly stochastic (P, P^T are both stochastic).

2. The convergence is exponential with rate given by $\rho(P)$ the second largest eigenvalue of P.

$$\boxed{\rho(P) = \max_{\lambda \in \Lambda(P) \setminus \{1\}} |\lambda|}$$

where $\Lambda(P)$ is the set of the eigenvalues of P.



Performance metrics

There are two types of problems:

1. Optimization problems: find the matrix P in a class which optimize the performance index.

2. Influence of the network topology: find how the network topology influences the the performance index.

We consider the second type of problems and more specifically we are interested in the influence of the number of nodes on the performance for the various types of network topologies.

Performance metrics

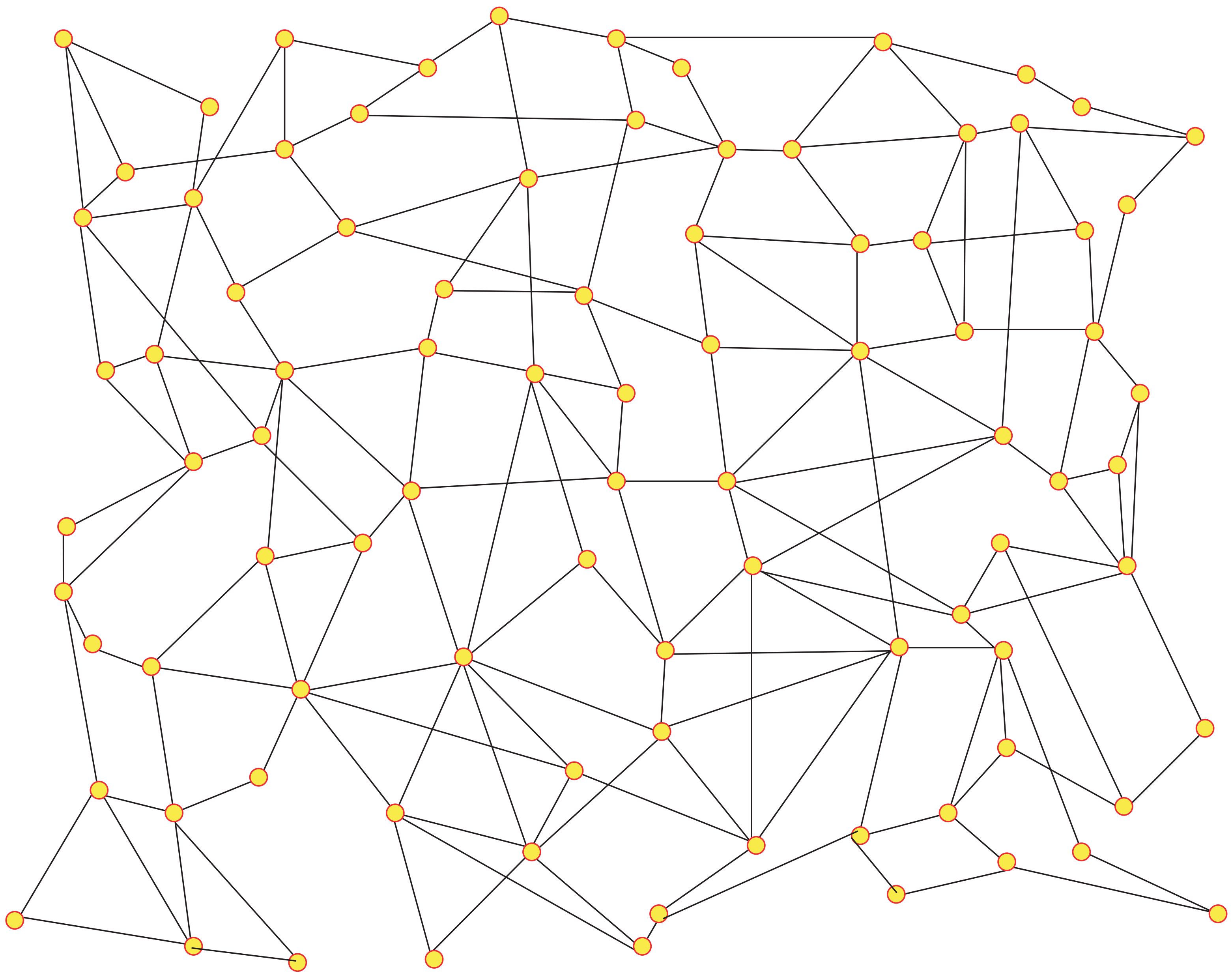
There is a huge literature which gives connections between the network topology and the performance index $\rho(P)$ *essential spectral radius*:

- Cheeger inequalities and expander graphs.
- Poincare' inequalities.
- Random graphs: Erdos-Renyi graphs, small world graphs, random geometric graphs.

These results are typically very hard and this performance metric is quite difficult to analyse.

Network topologies

We consider here network topologies coming from wireless sensor networks applications, namely the geometric graphs



Network topologies

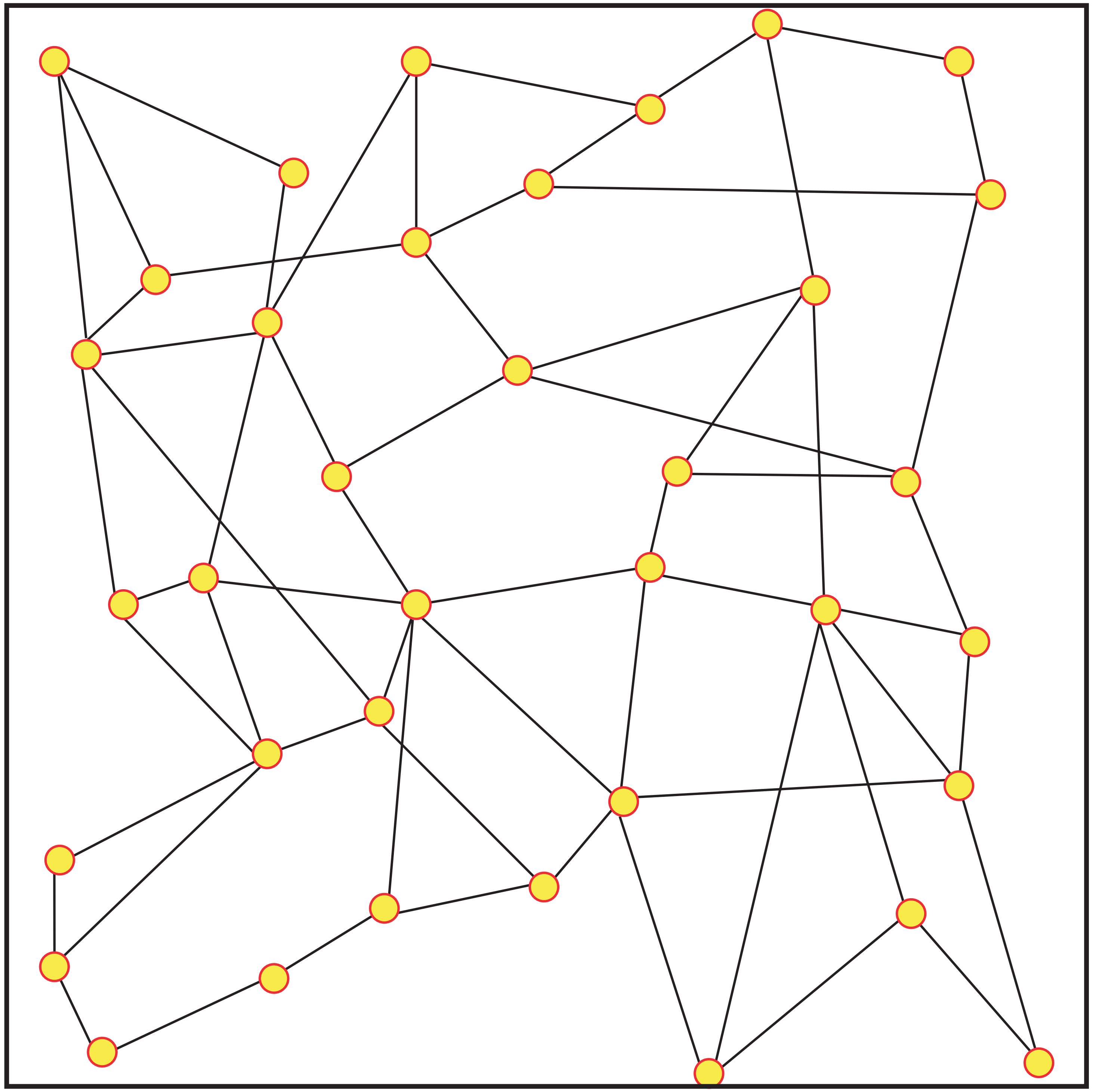
Formal definition of geometric graphs (Doyle-Snell, Barooah-Hespanha)

A geometric graph is a graph deployed in $\mathbb{R}^d$ characterized by five parameters:

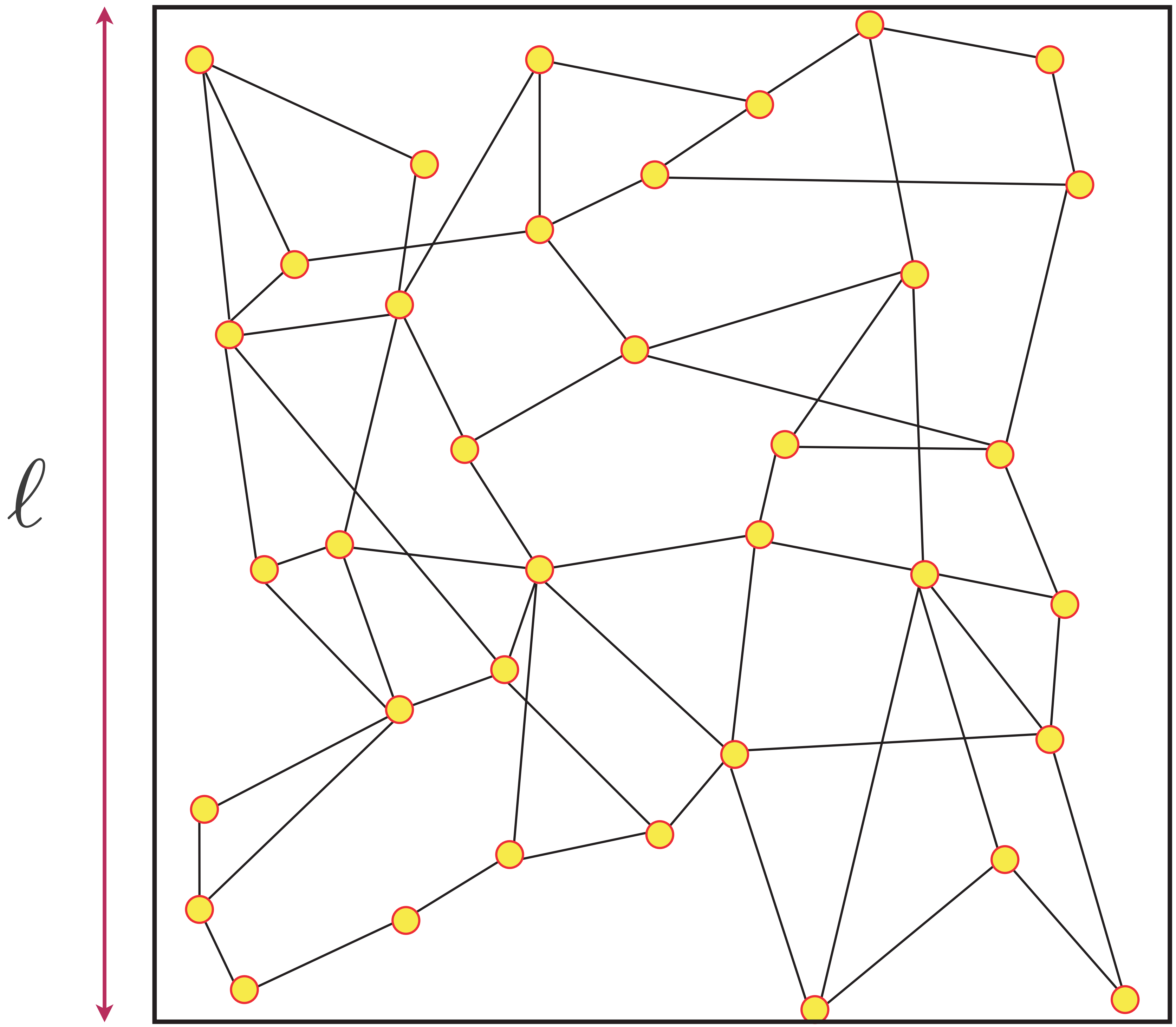
- The edge length ℓ of an hypercube in which all the nodes lye.
- The minimum Euclidean distance s between two nodes.
- The maximum range of communication r.
- The radius γ of the largest empty ball.
- The minimum ρ of the ratios between the graphical and the Euclidean distance between nodes.

Examples: perturbed grids, random geometric graphs whp.

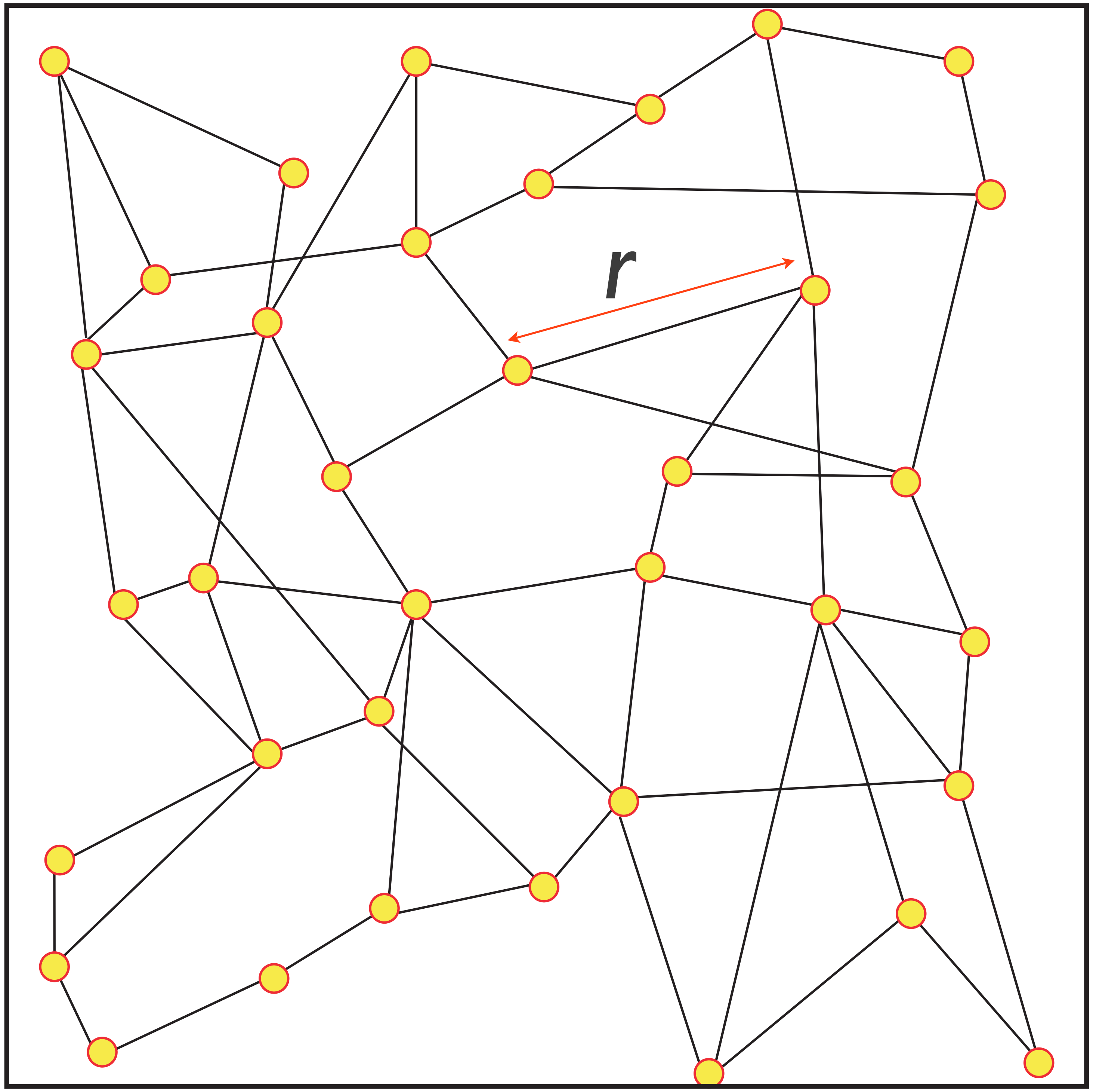
Network topologies



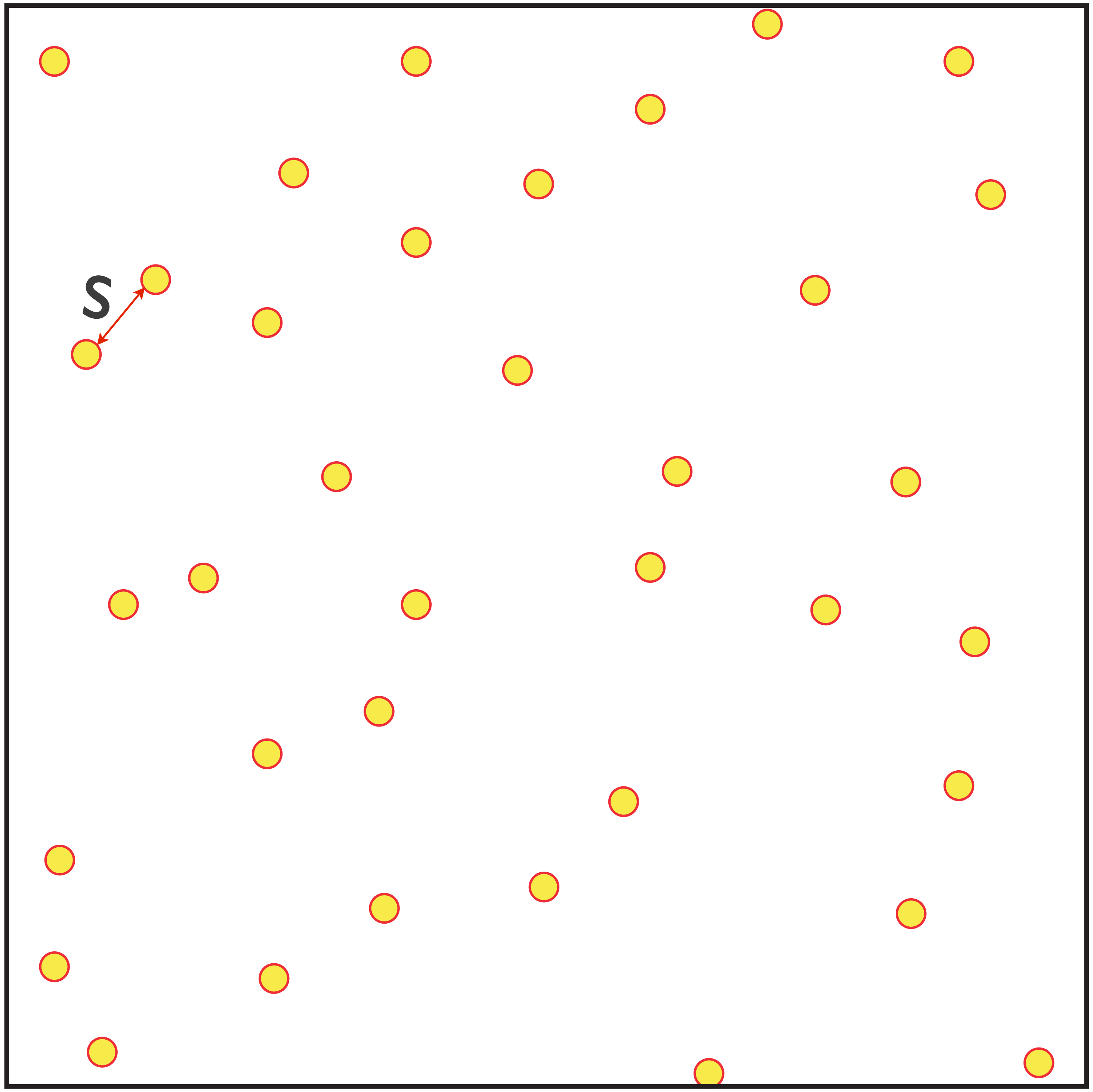
Network topologies



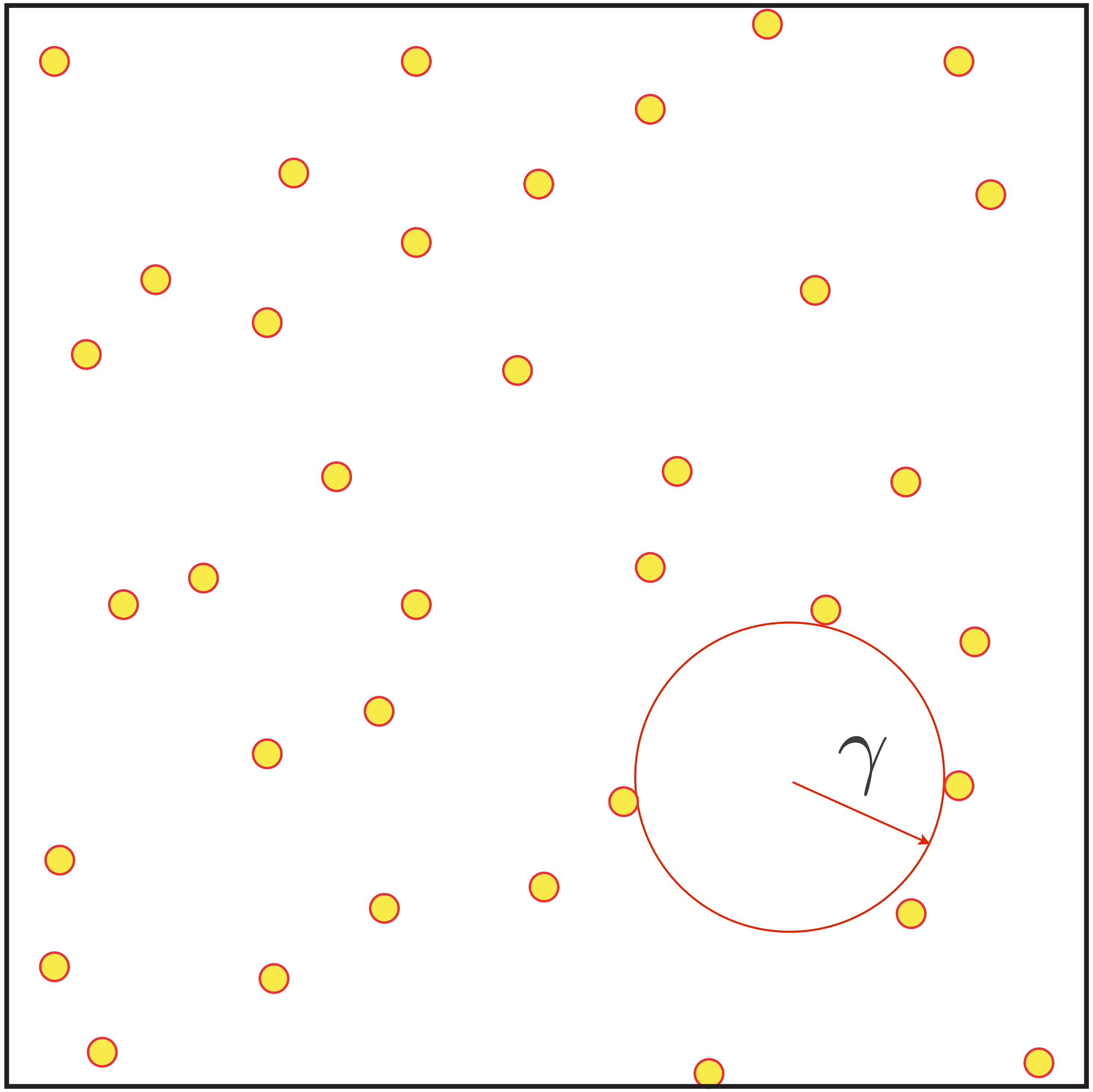
Network topologies



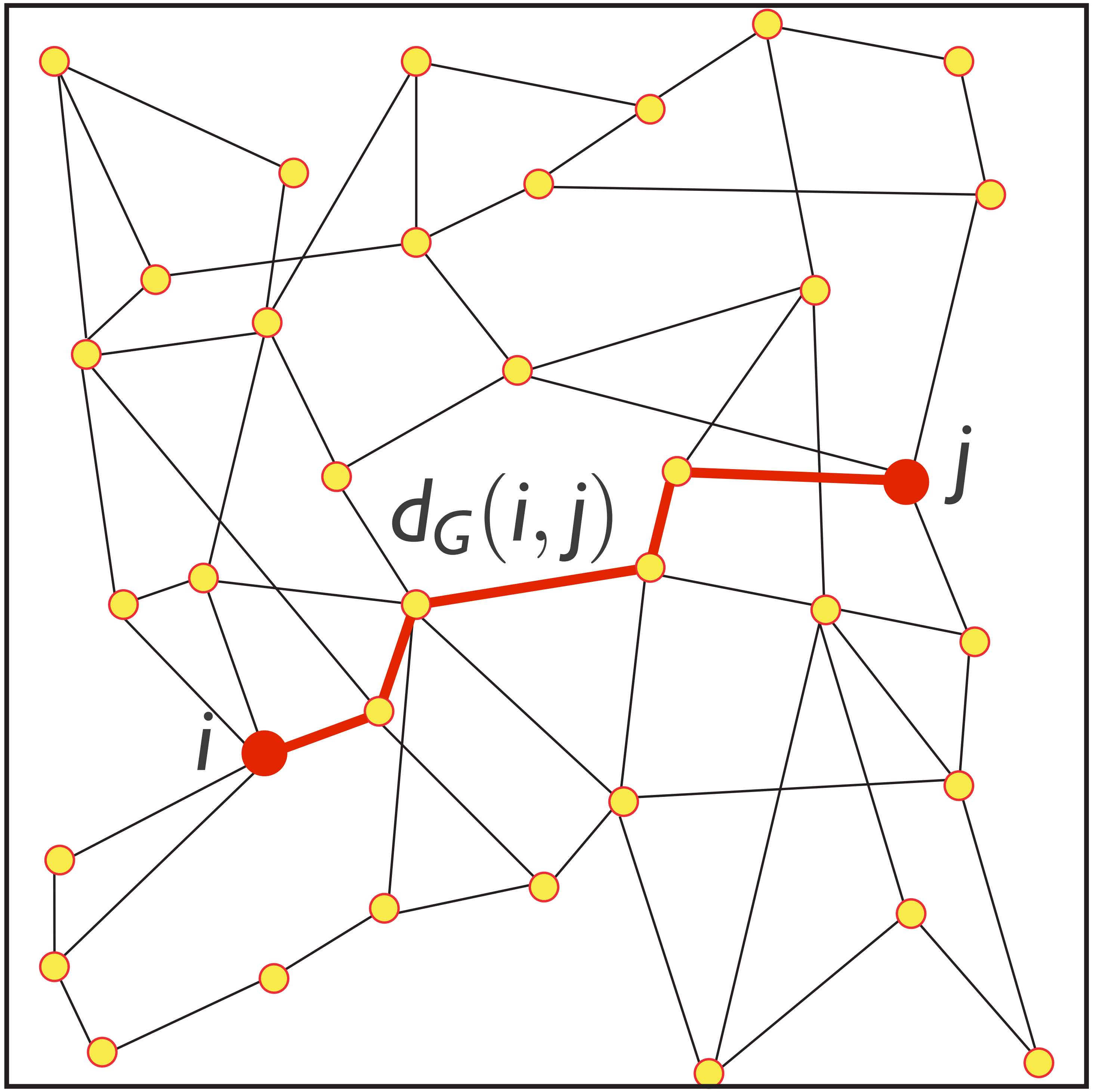
Network topologies



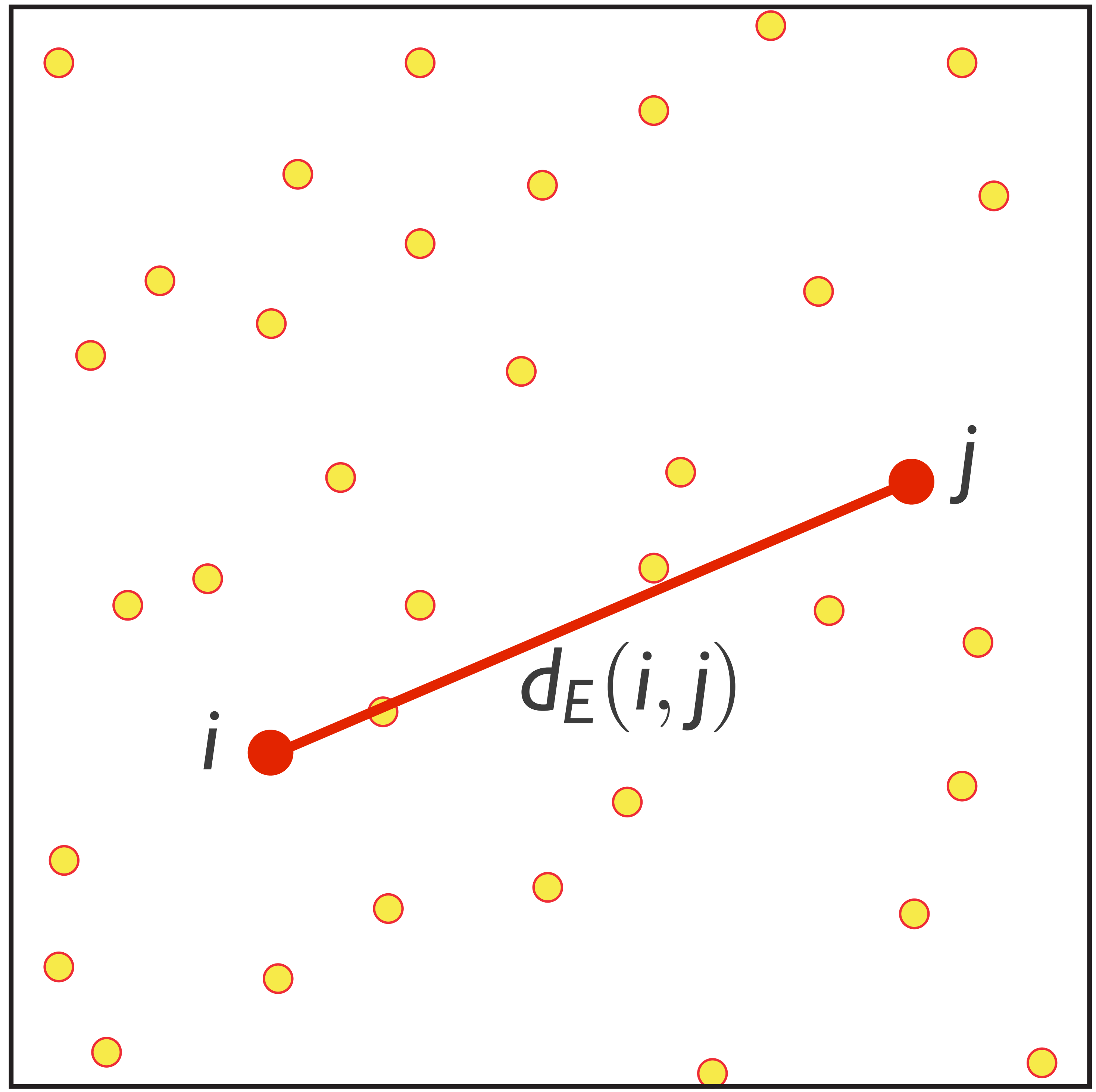
Network topologies



Network topologies



Network topologies

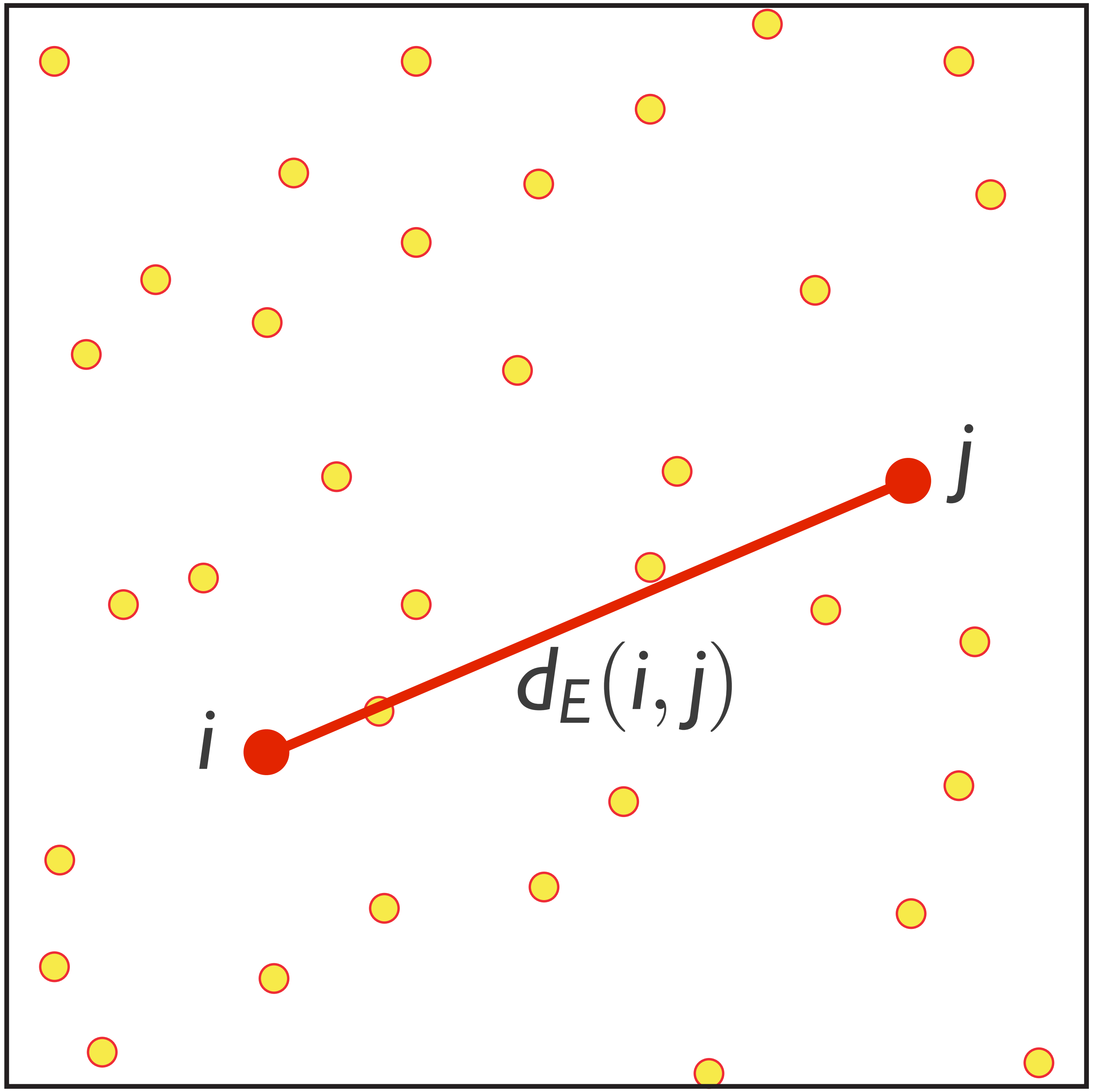


Network topologies

$$\rho = \min \left\{ \frac{d_E(i,j)}{d_G(i,j)} \right\}$$

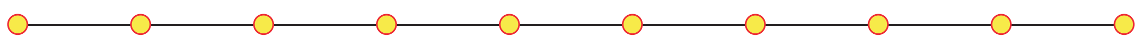
$$\Downarrow$$

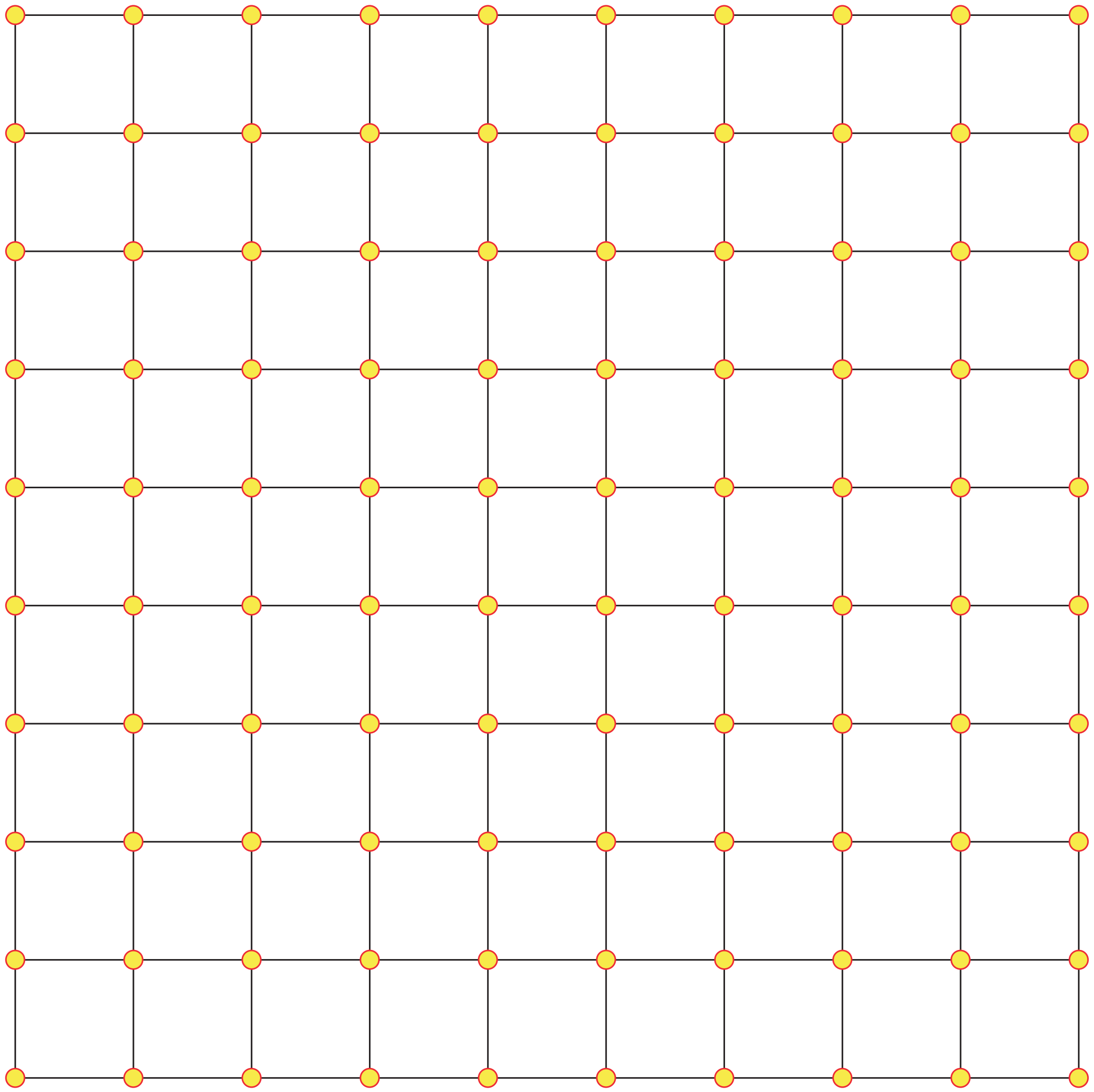
$$d_G(i,j) \leq \frac{d_E(i,j)}{\rho}$$



Network topologies

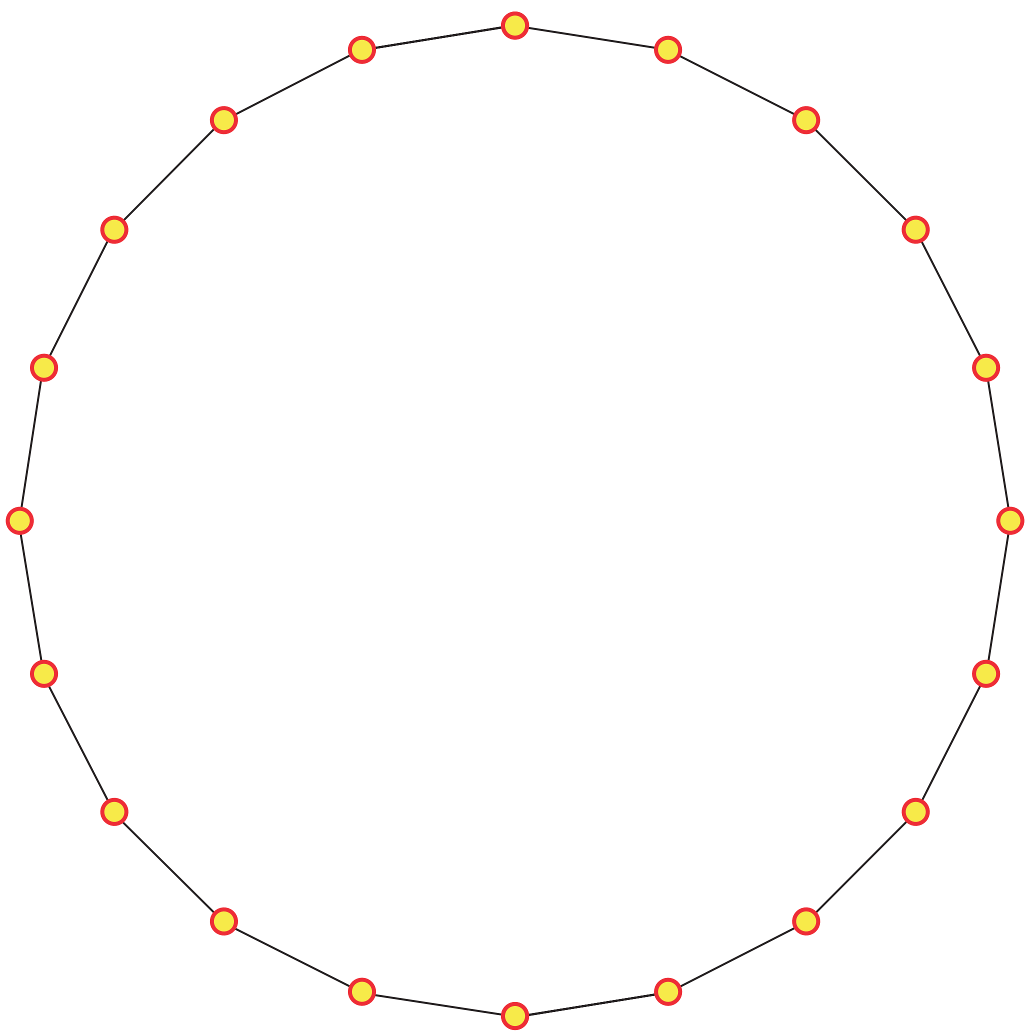
In certain cases we need to restrict to lines and grids

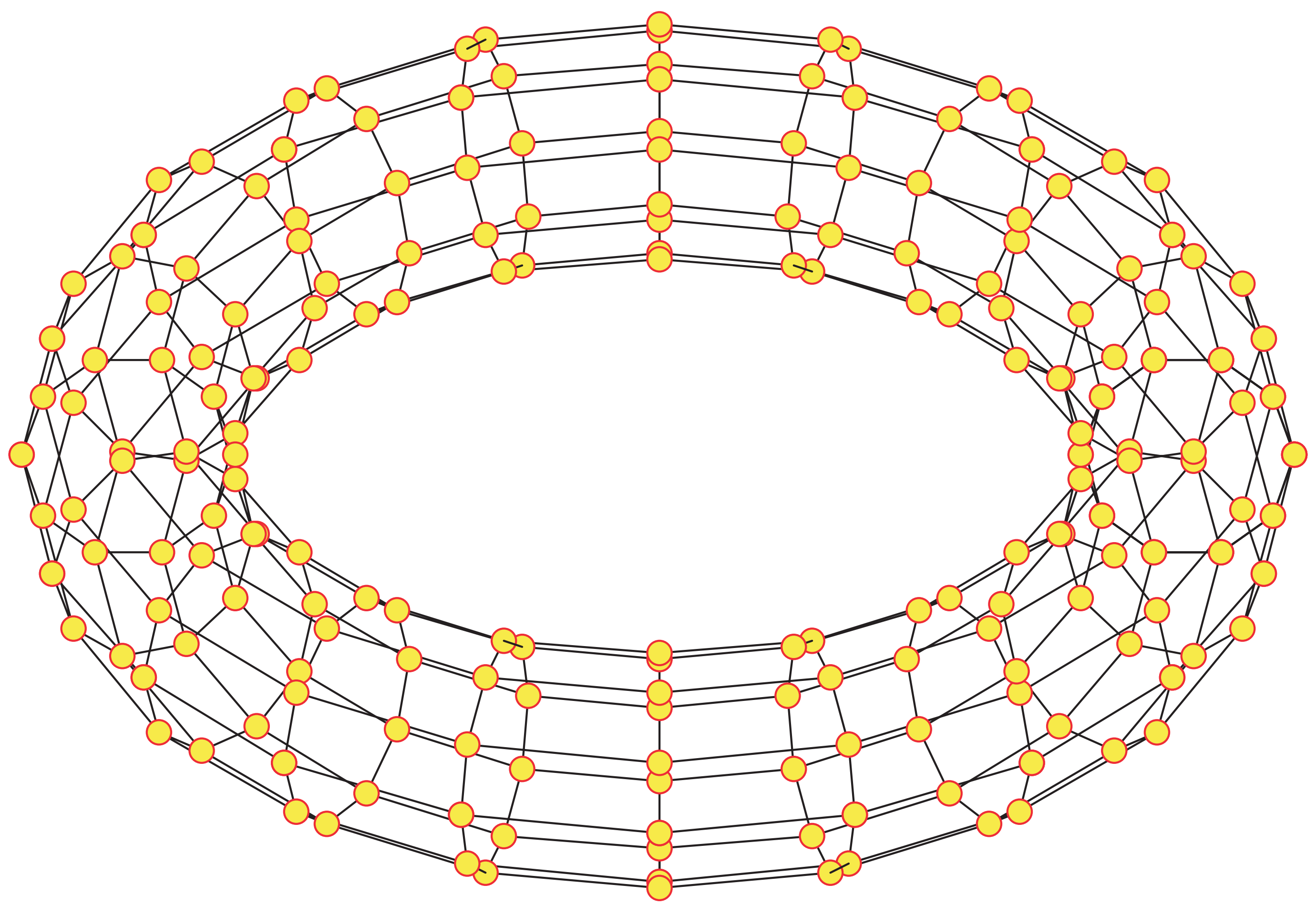




Network topologies

The simplest “geometric” graphs are circles and toruses (no boundary effects)





Rate of convergence

THEOREM (Boyd, Ghosh, Prabhakar, Sha 2006, Lovisari, Zampieri 2011)

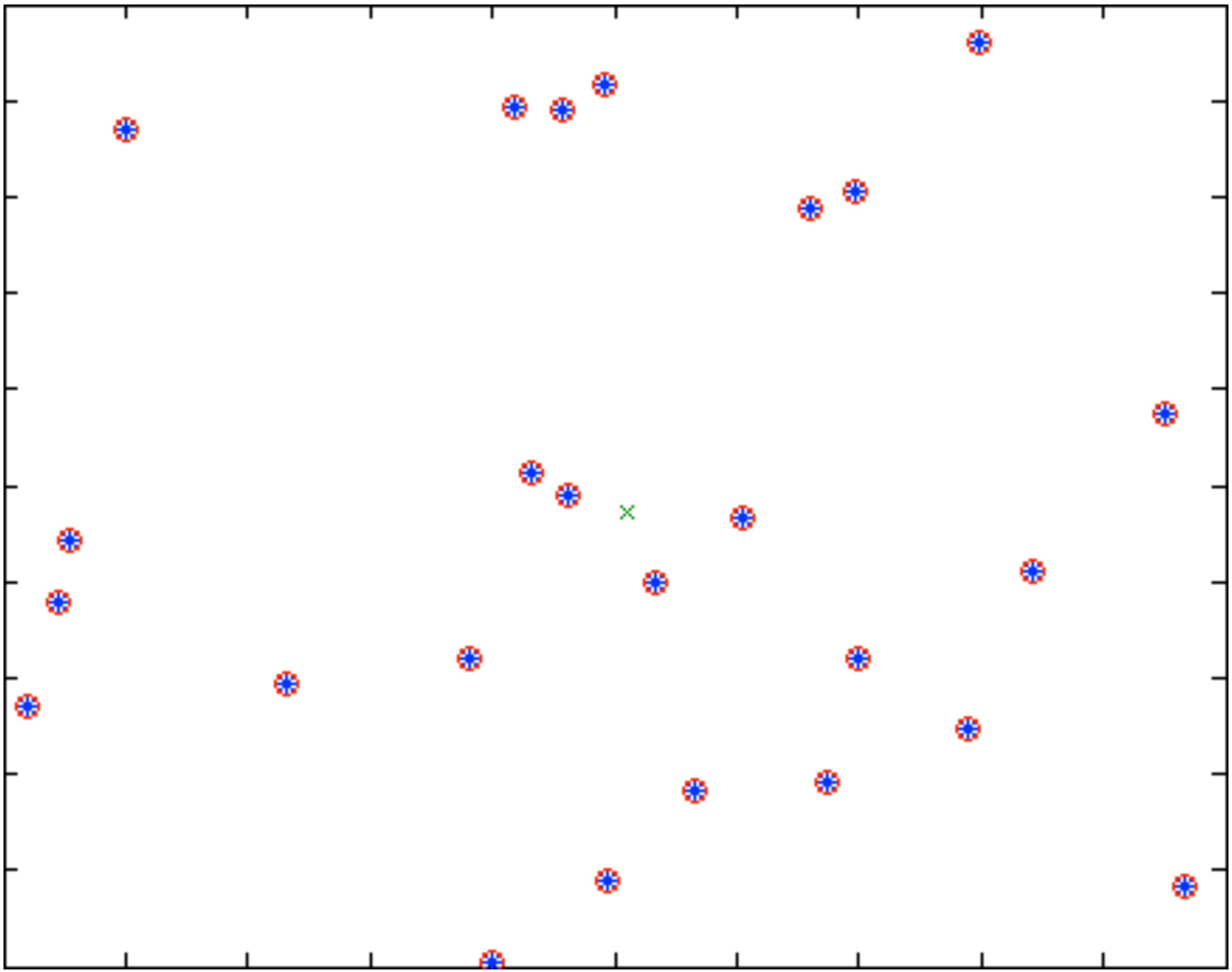
Let $\mathcal{G}$ be a d-dimensional connected geometric graph/grid/torus and let P be any stochastic matrix compatible with $\mathcal{G}$. Then

$$\boxed{1 - \frac{C'}{N^{2/d}} \leq \rho(P) \leq 1 - \frac{C''}{N^{2/d}}}$$

where N is the number of nodes and C', C'' are suitable constants.

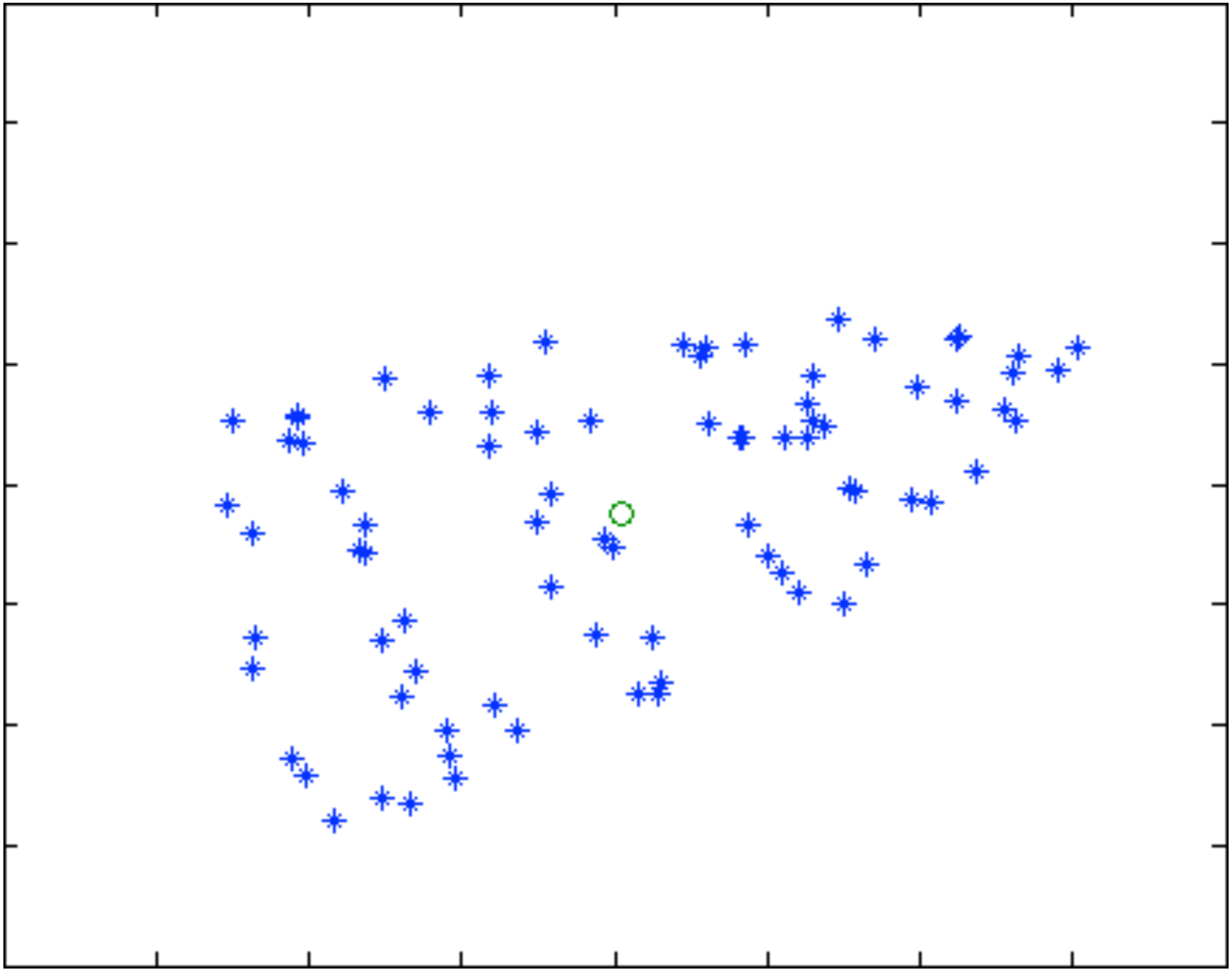
Rate of convergence

Rate of convergence

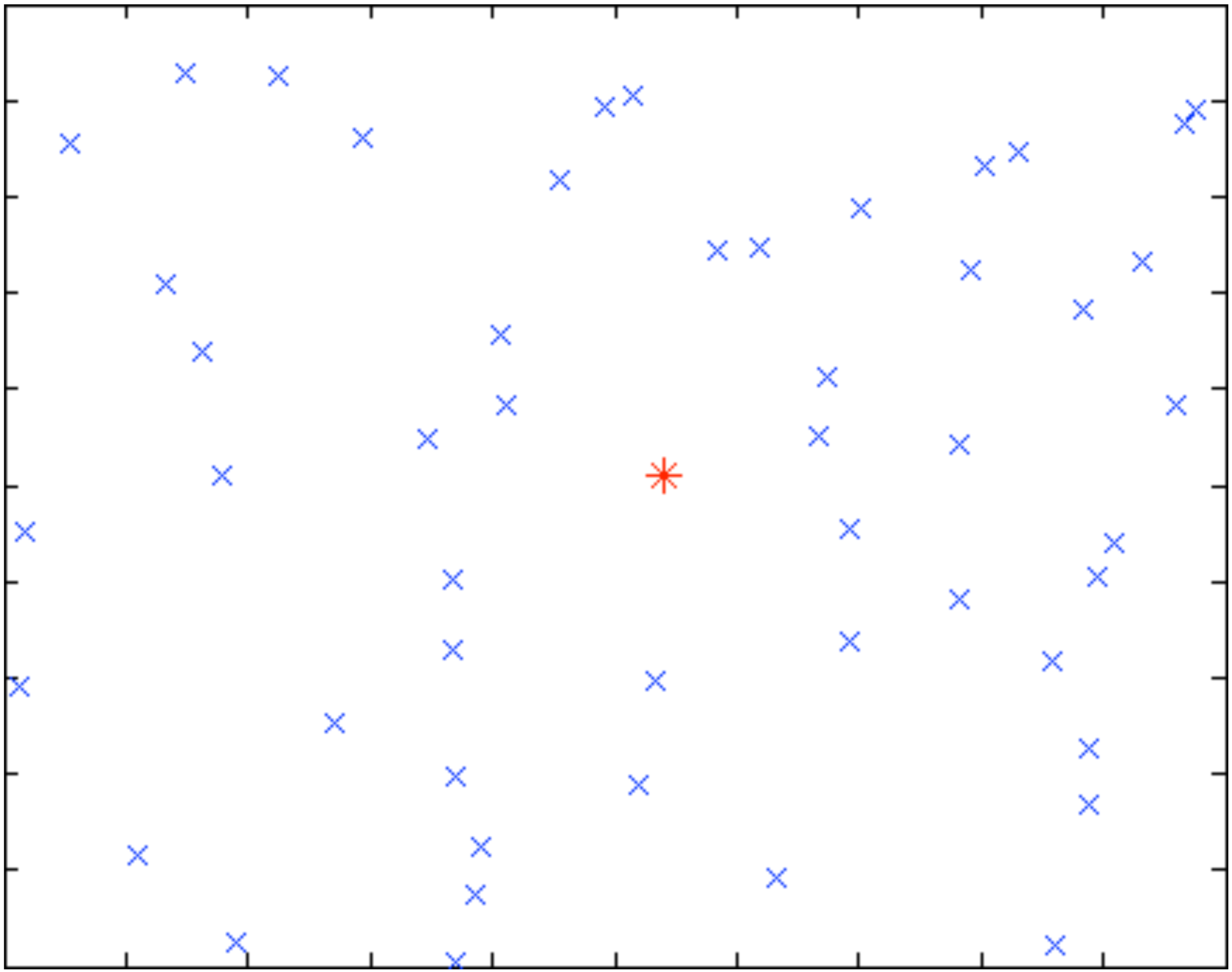


Rate of convergence

Rate of convergence

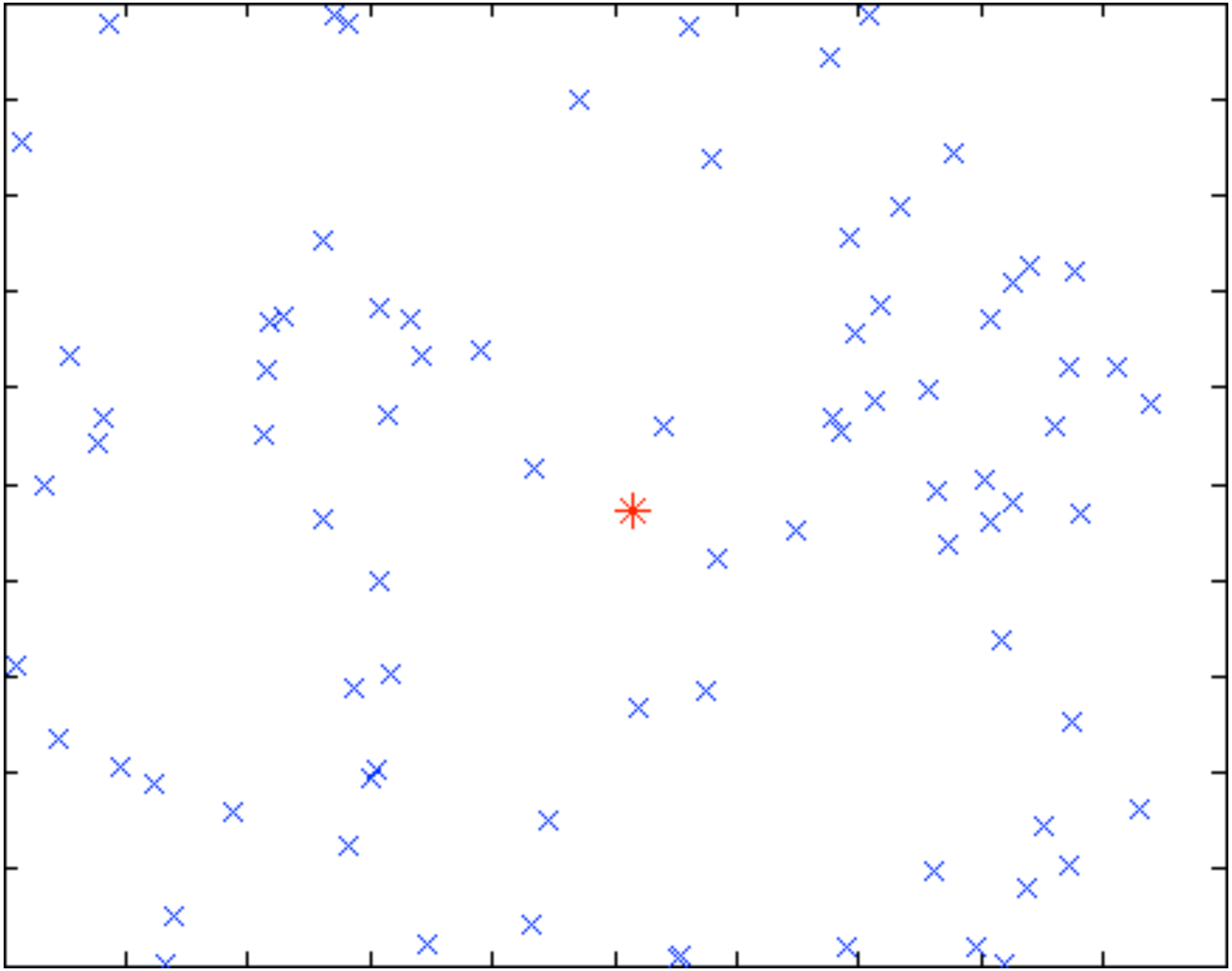


Rate of convergence



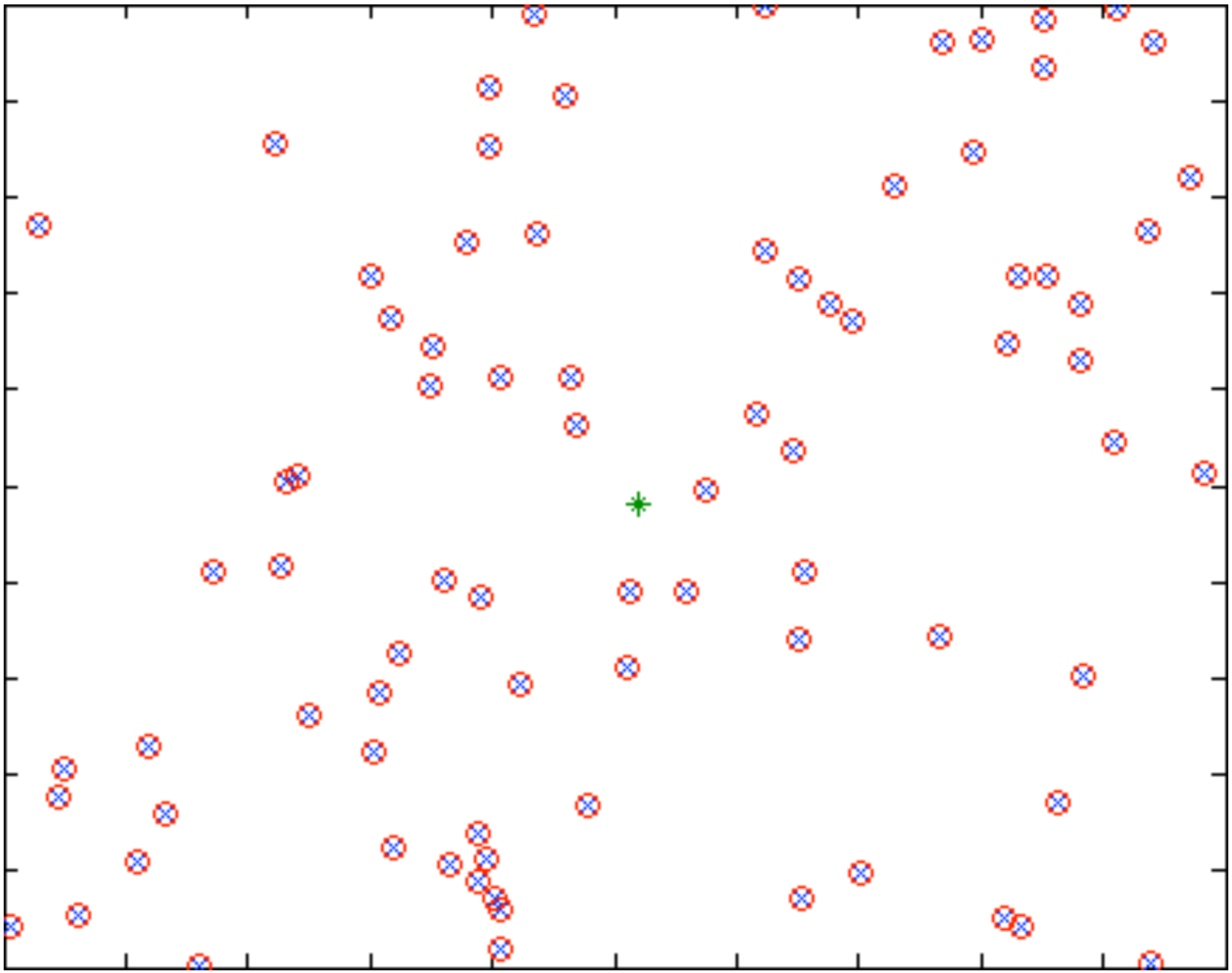
Circle

Rate of convergence



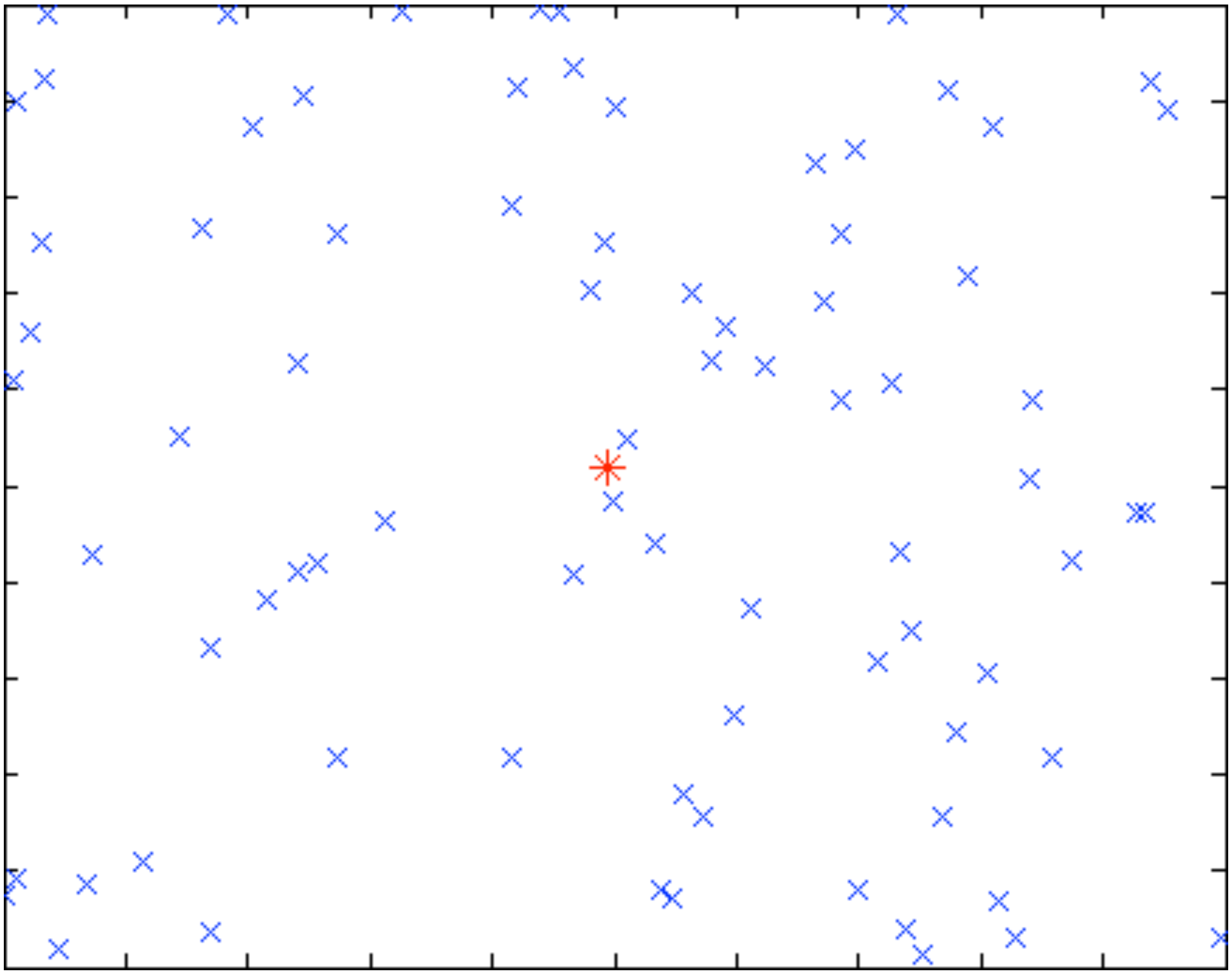
Torus

Rate of convergence



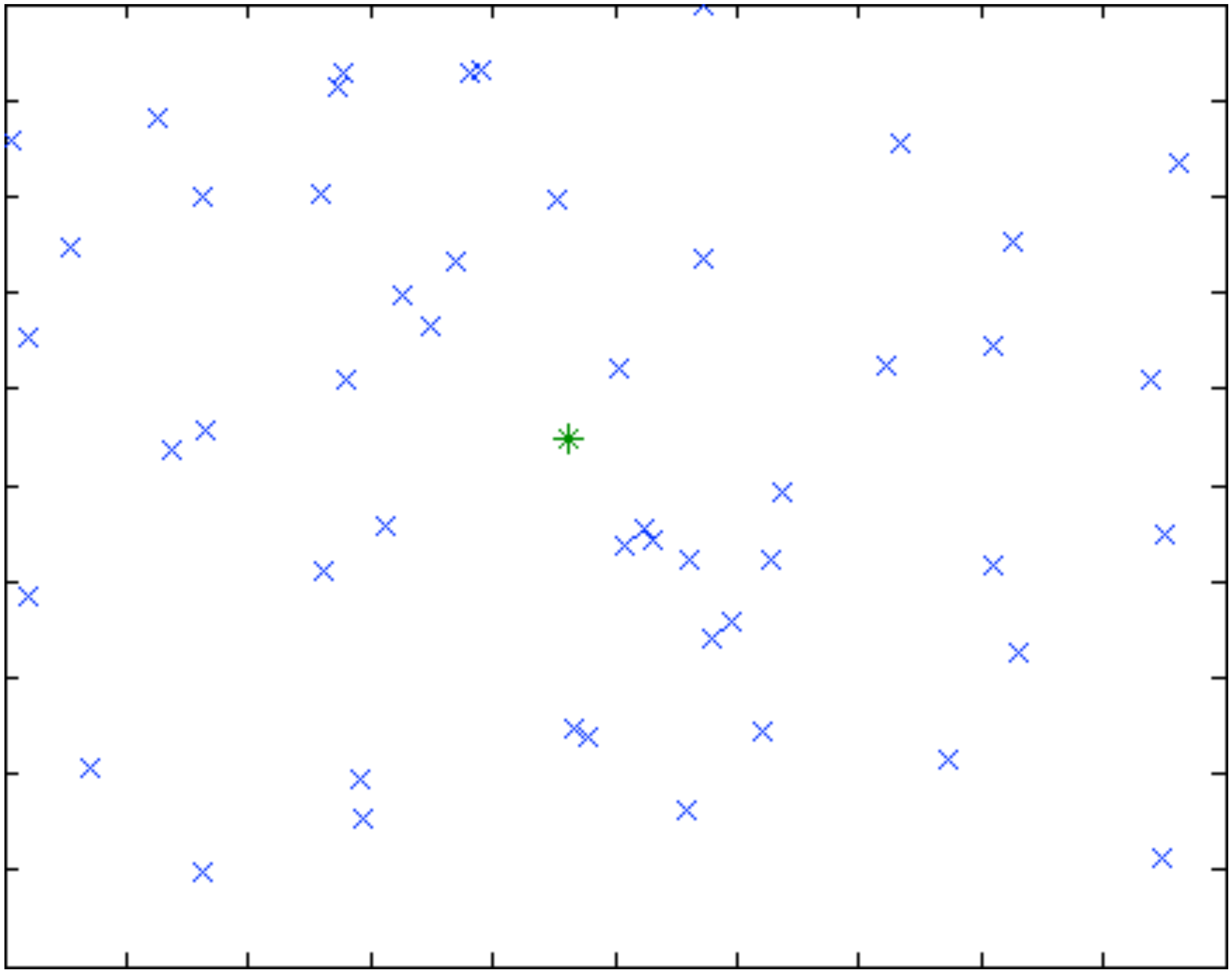
Circle and torus

Rate of convergence



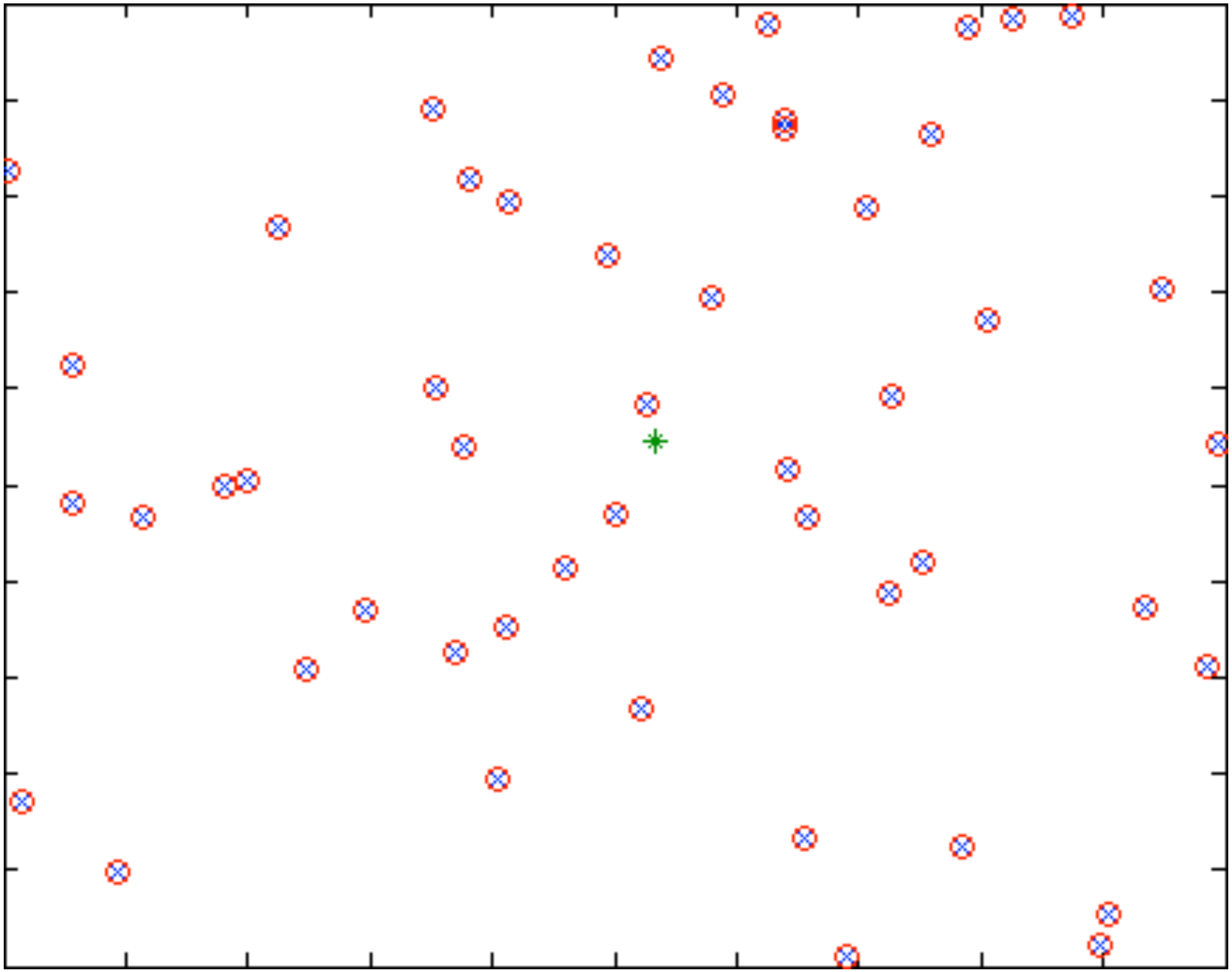
Geometric graph

Rate of convergence



Random graph

Rate of convergence



Random graph and circle

H^2 performance

Assume that the initial conditions are i.i.d. random variables $x_i(0)$ and consider the following index

$$\boxed{J(P) := \frac{1}{N} E\left[\sum_{t=0}^{\infty} ||x(t) - x(\infty)||^2\right]}$$

where $||\cdot||$ is the 2-norm.

H^2 performance

The same cost appears in a different context (Xiao, Boyd), (Bamieh, Jovanovic, Mitra, Patterson) and (Carli, Frasca, Fagnani, Zampieri)

Assume that the standard consensus algorithm is corrupted by and i.i.d. noise

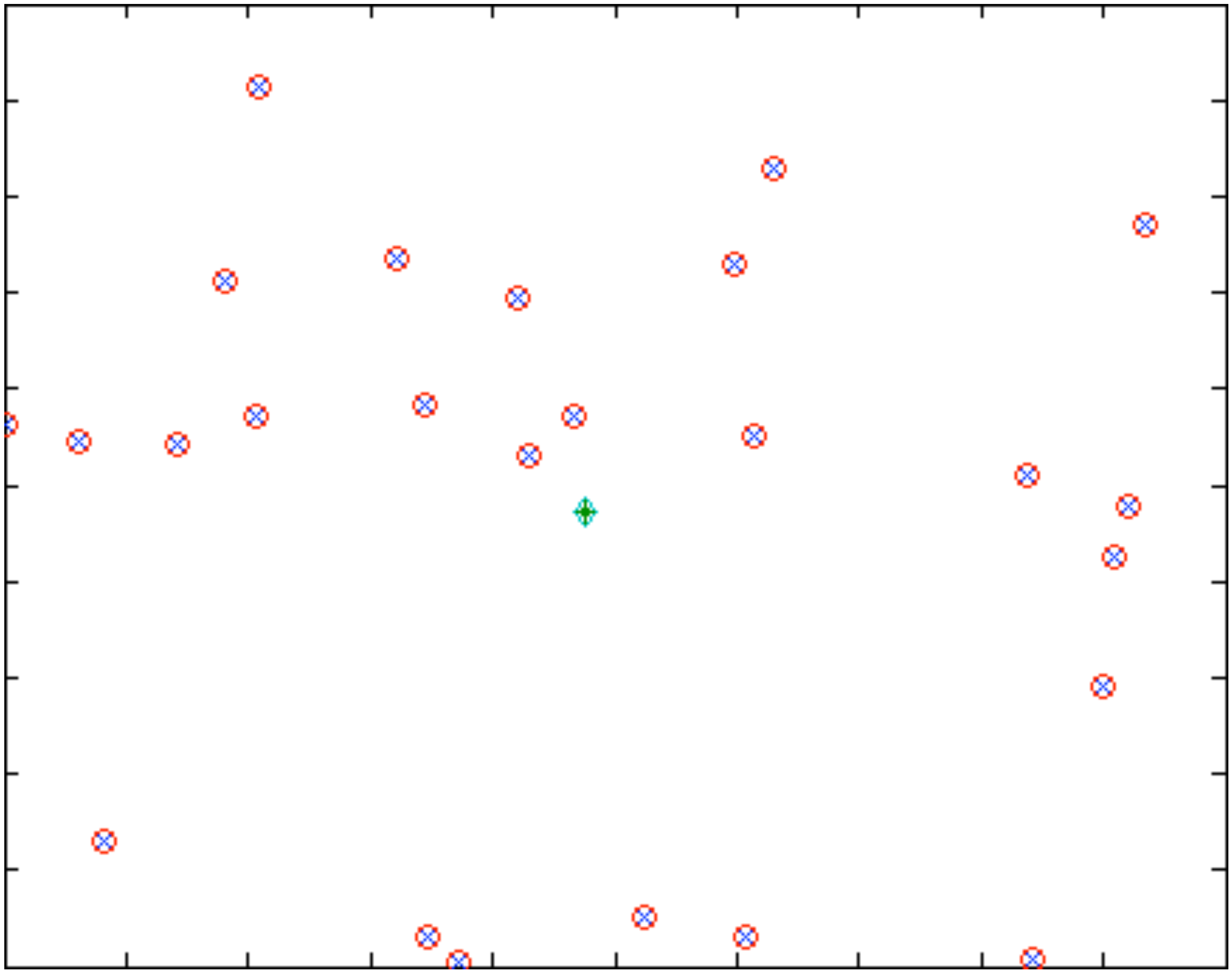
$$x(t+1) = Px(t) + n(t)$$

and consider the following index

$$\boxed{J(P) := \limsup_{t\to\infty} \frac{1}{N} E\left\{||x(t) - x_A(t)\mathbb{1}||\right\}}$$

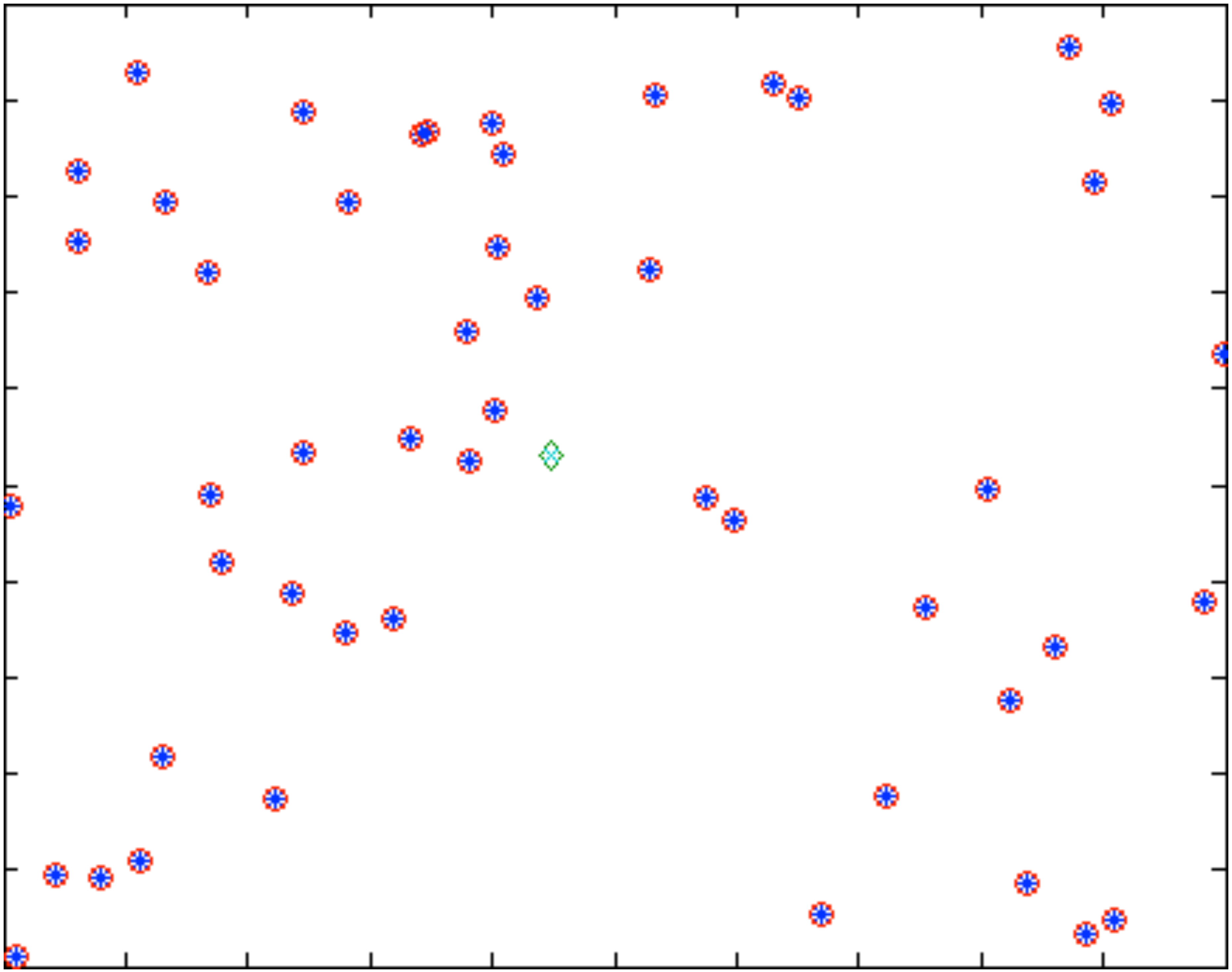
where $x_A(t) = 1/N \sum x_i(t)$ is is the average of the components of $x(t)$ and $\mathbb{1}$ is the column vector with entries equal to 1.

Consensus with noise



Circle

Consensus with noise



Geometric graph

H^2 performance

Assume for simplicity that P is symmetric. Then

$$
\begin{aligned}
J(P) &= \frac{1}{N}\operatorname{tr}\sum_{t=0}^{\infty}\left(P^{2t}-\frac{1}{N}\mathbb{1}\mathbb{1}^T\right)\\
&= \frac{1}{N}\sum_{\lambda\in\Lambda(P)\setminus\{1\}}\frac{1}{1-\lambda^2}
\end{aligned}
$$

where $\mathbb{1}$ is the column vector with all entries equal to 1.

L^2 performance

THEOREM

Let $\mathcal{G}$ be a d-dimensional connected geometric graph/grid/torus and let P be any symmetric stochastic matrix compatible with $\mathcal{G}$. Then

$$\boxed{\begin{array}{ll} C'N \leq J(P) \leq C''N & \text{if } d = 1 \\ C' \log(N) \leq J(P) \leq C'' \log(N) & \text{if } d = 2 \\ C' \leq J(P) \leq C'' & \text{if } d \geq 3 \end{array}}$$

where N is the number of nodes and C', C'' are suitable constants.

Idea of the proof

Idea of the proof

$Z := P^2$ symmetric stochastic matrix

Electrical network (Doyle, Snell)

Idea of the proof

$$Z := P^2 \text{ symmetric stochastic matrix}$$

$$\Downarrow$$

Electrical network with N nodes and conductance $Z_{uv}/2$ between the nodes u and v (namely resistance $2/Z_{uv}$).

Electrical network (Doyle, Snell)

Idea of the proof

$Z := P^2$ symmetric stochastic matrix

Electrical network with N nodes and conductance $Z_{uv}/2$ between the nodes u and v (namely resistance $2/Z_{uv}$).

$\mathcal{R}_{uv}$ is the resistance of the electric network between the node u and the node v.

Electrical network (Doyle, Snell)

Idea of the proof

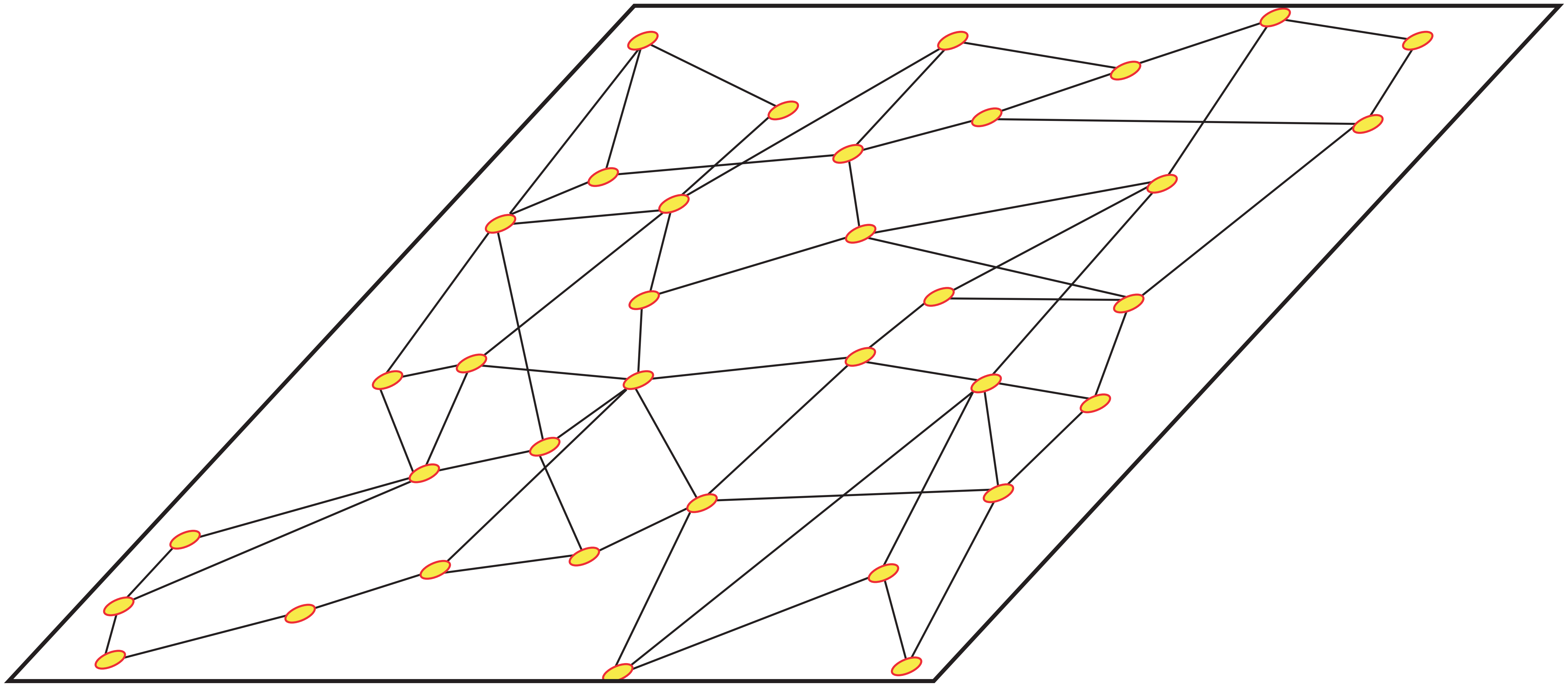
$Z := P^2$ symmetric stochastic matrix

Electrical network with N nodes and conductance $Z_{uv}/2$ between the nodes u and v (namely resistance $2/Z_{uv}$).

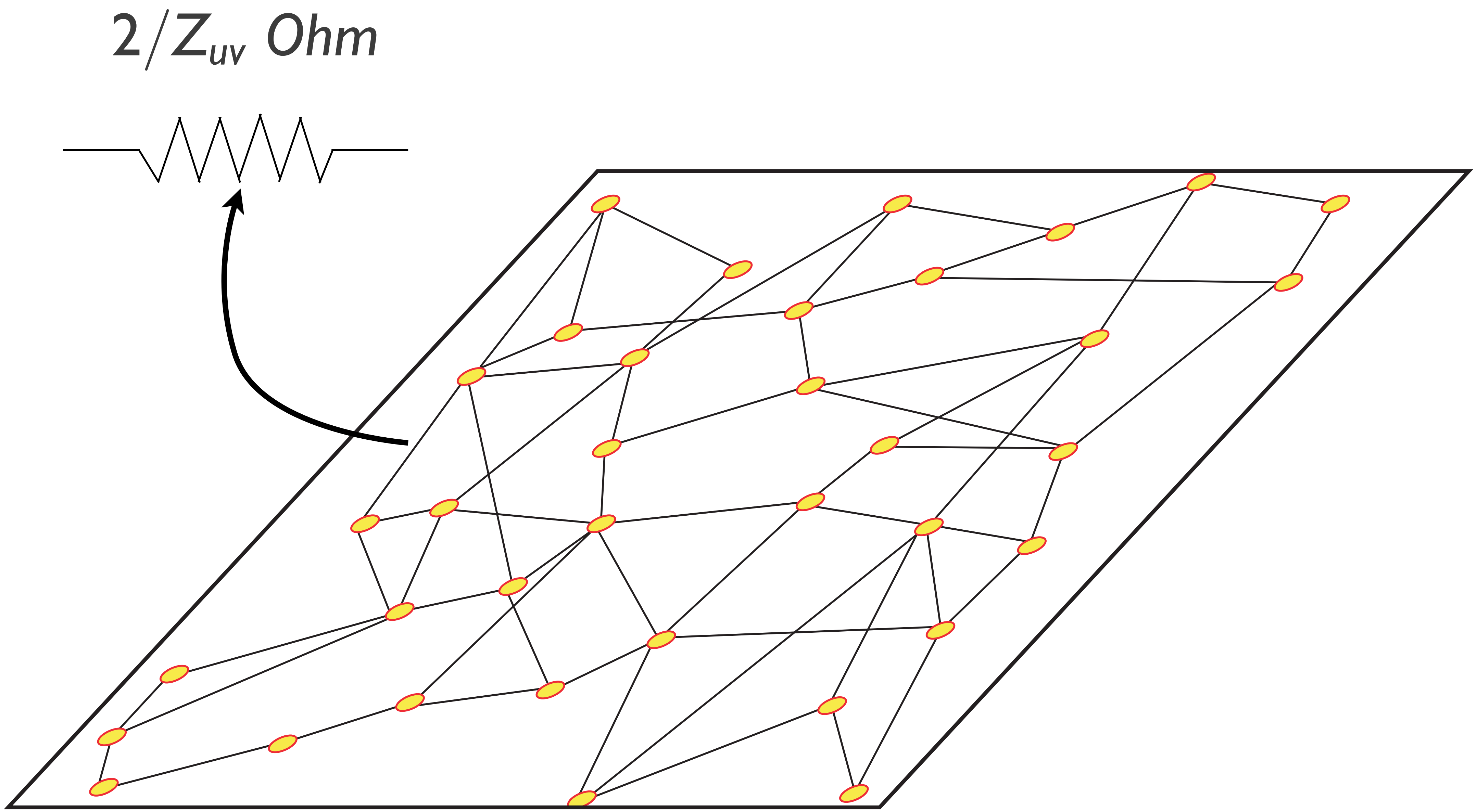
$\mathcal{R}_{uv}$ is the resistance of the electric network between the node u and the node v.

$$\boxed{J(P) = \frac{1}{N^2} \sum_{u \neq v} \mathcal{R}_{uv}}$$

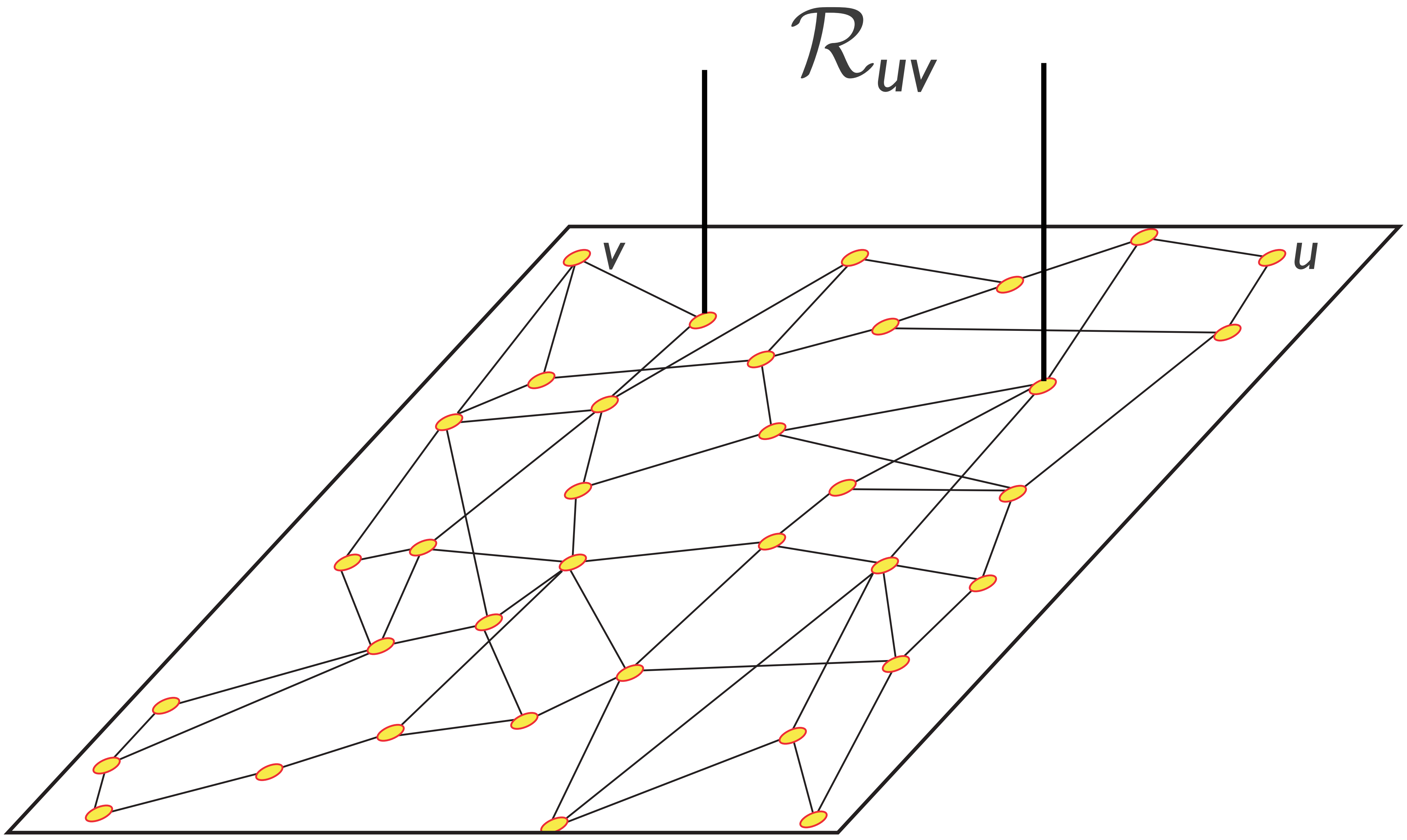
Electrical network (Doyle, Snell)



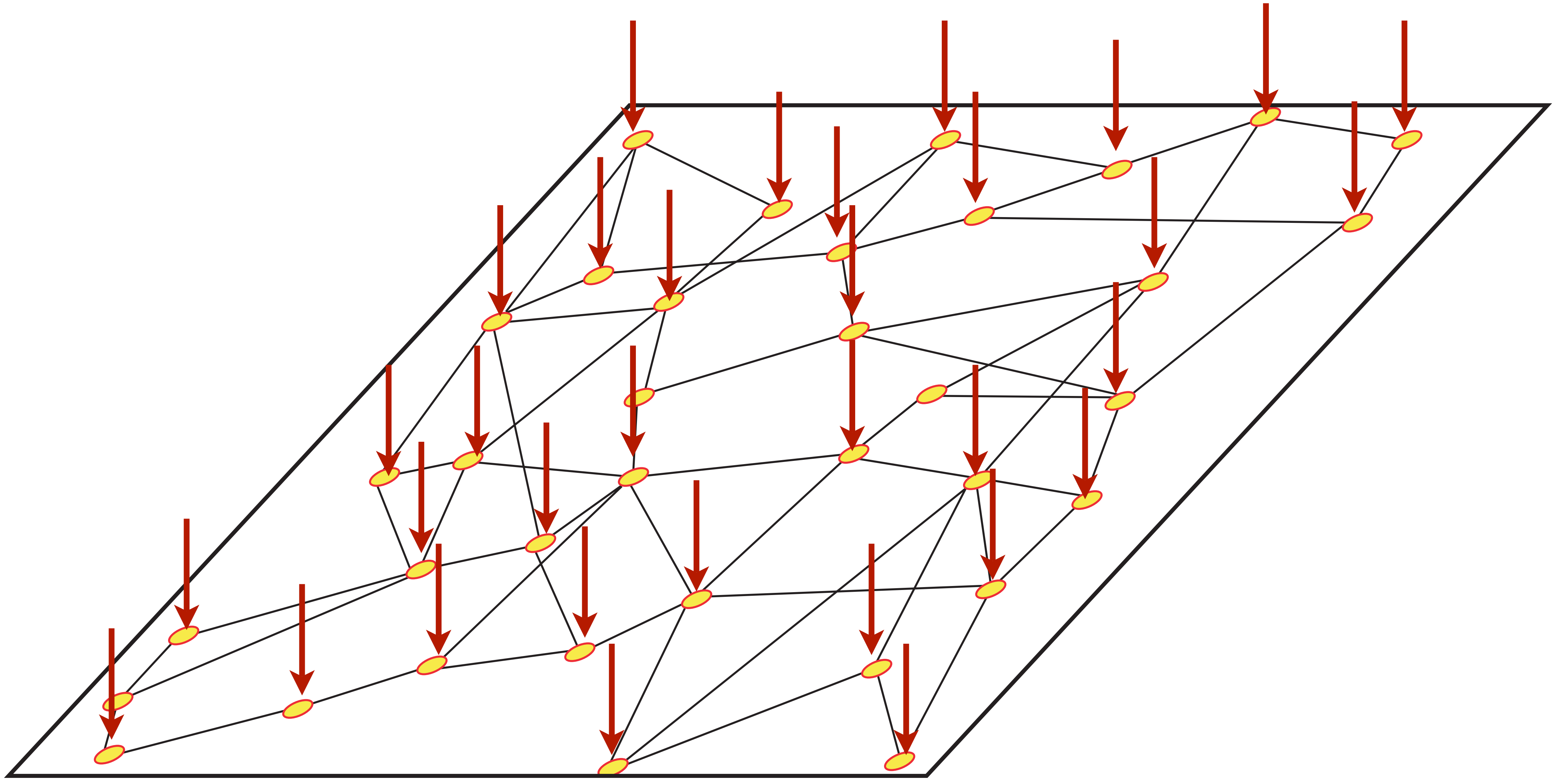
Electrical network (Doyle, Snell)



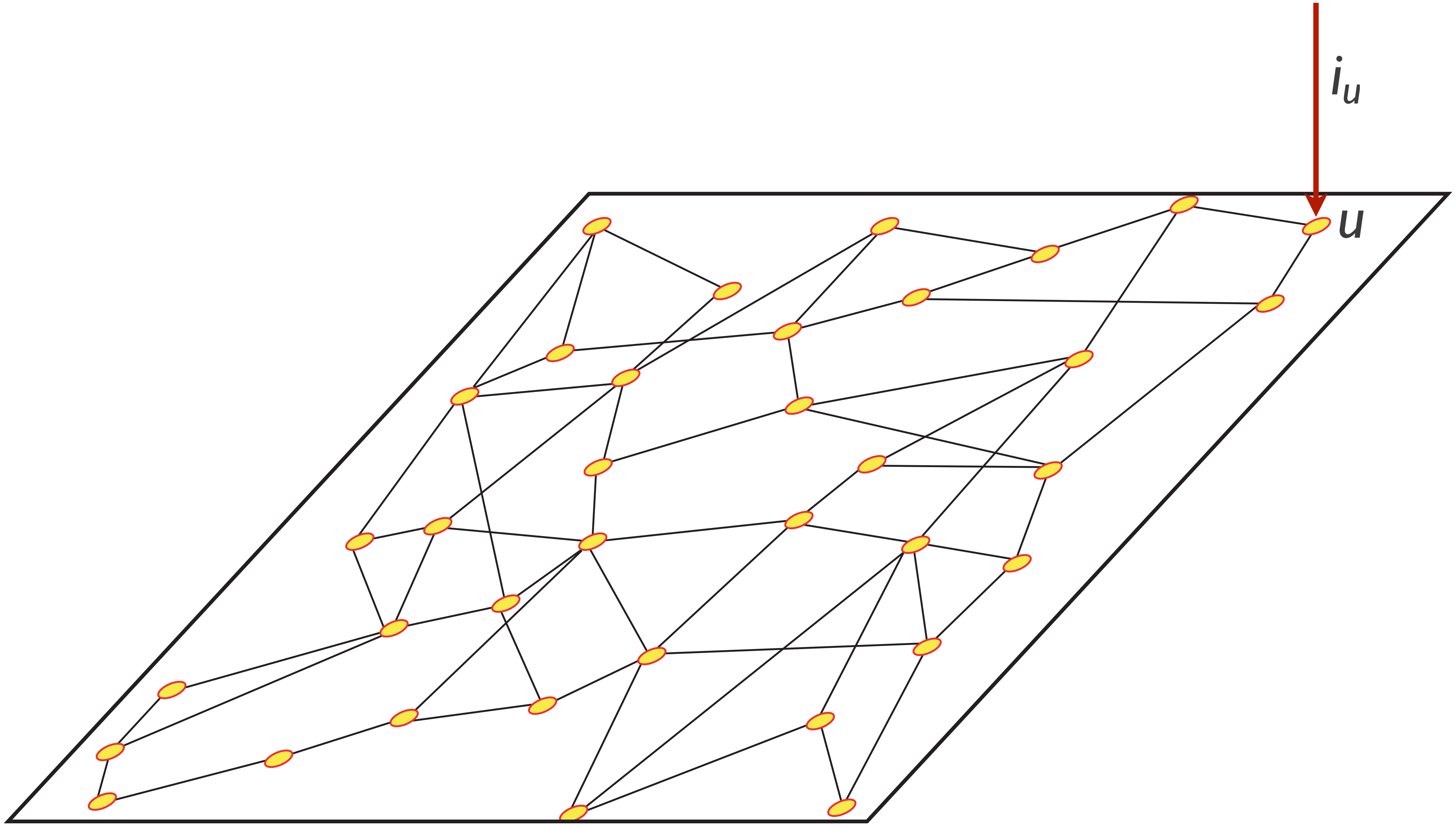
Electrical network (Doyle, Snell)



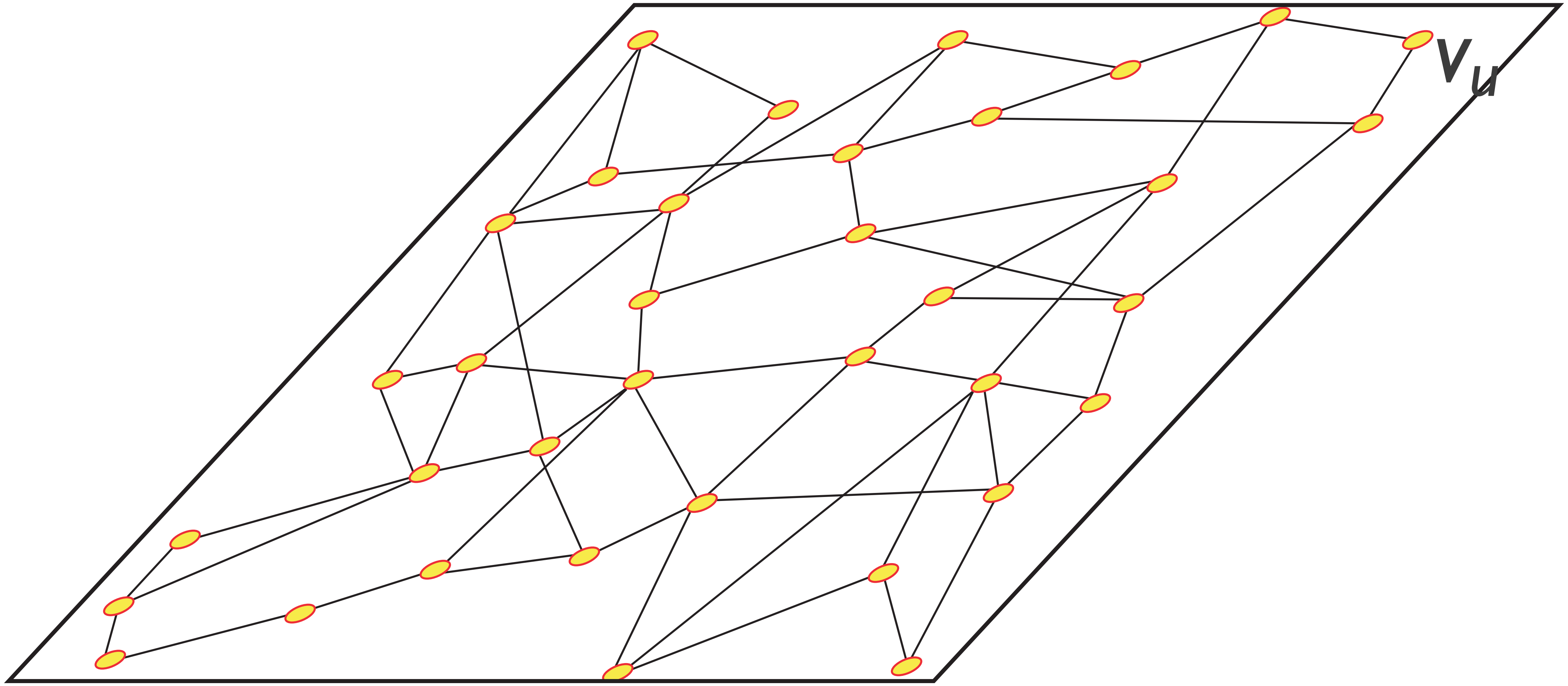
Electrical network (Doyle, Snell)



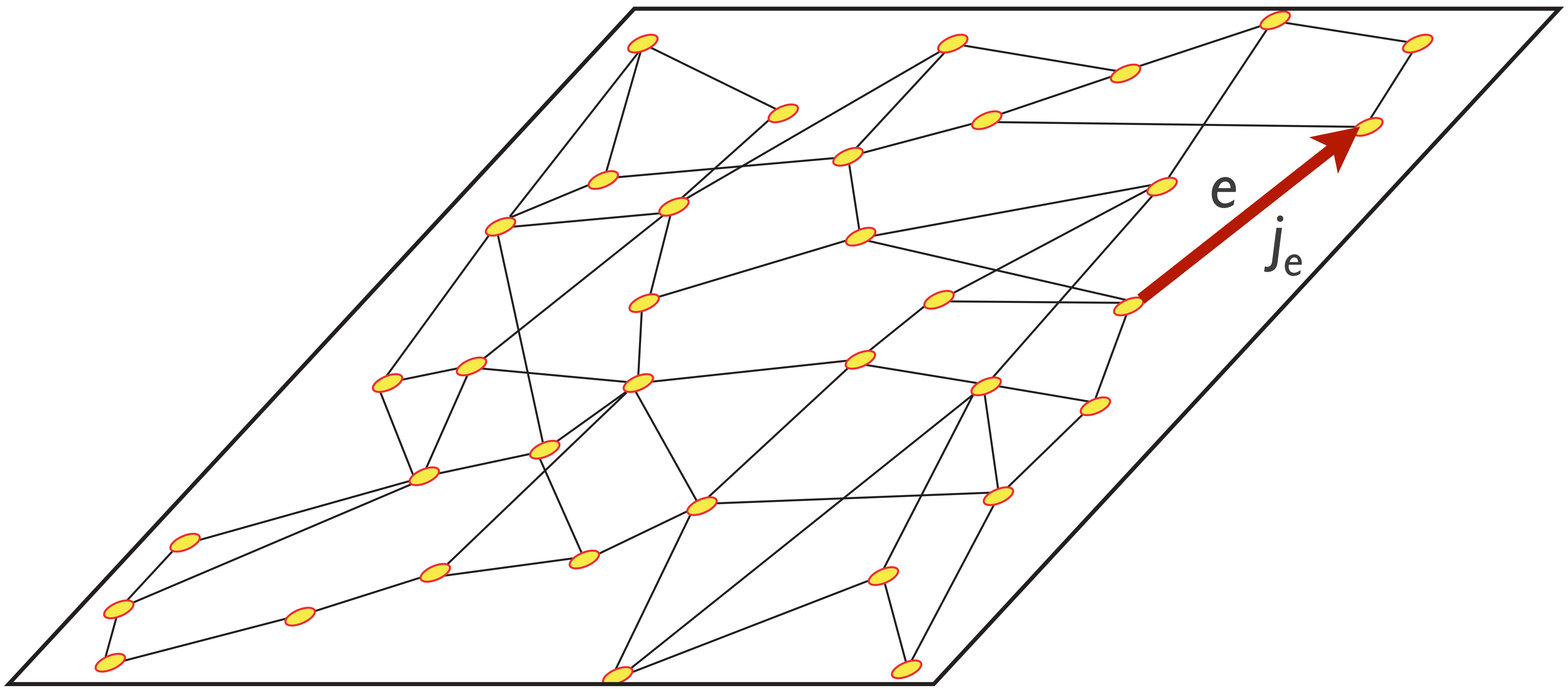
Electrical network (Doyle, Snell)



Electrical network (Doyle, Snell)



Electrical network (Doyle, Snell)



Electrical network (Doyle, Snell)

The potential v can be found by solving the equations

$$\begin{cases} A^T j = i & \text{Current Kirchhoff's current law} \\ \mathcal{C} A v = j & \text{Ohm's law} \\ \mathbb{1}^T v = 0 & \text{Condition to get uniqueness} \end{cases}$$

where A is the incidence matrix of the graph and $\mathcal{C}$ is a diagonal matrix having $Z_{ij}/2$ as diagonal elements. This equation is equivalent to the following

$$\begin{bmatrix} A^T \mathcal{C} A & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} i \\ 0 \end{bmatrix}$$

and so

$$\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} A^T \mathcal{C} A & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} i \\ 0 \end{bmatrix}$$

Electrical network (Doyle, Snell)

$$\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} A^T \mathcal{C} A & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} i \\ 0 \end{bmatrix}$$

Notice that

$$\boxed{A^T \mathcal{C} A = I - Z}$$

It turns out that

$$\begin{bmatrix} I - Z & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} G(Z) & N^{-1}\mathbb{1} \\ N^{-1}\mathbb{1}^T & 0 \end{bmatrix}$$

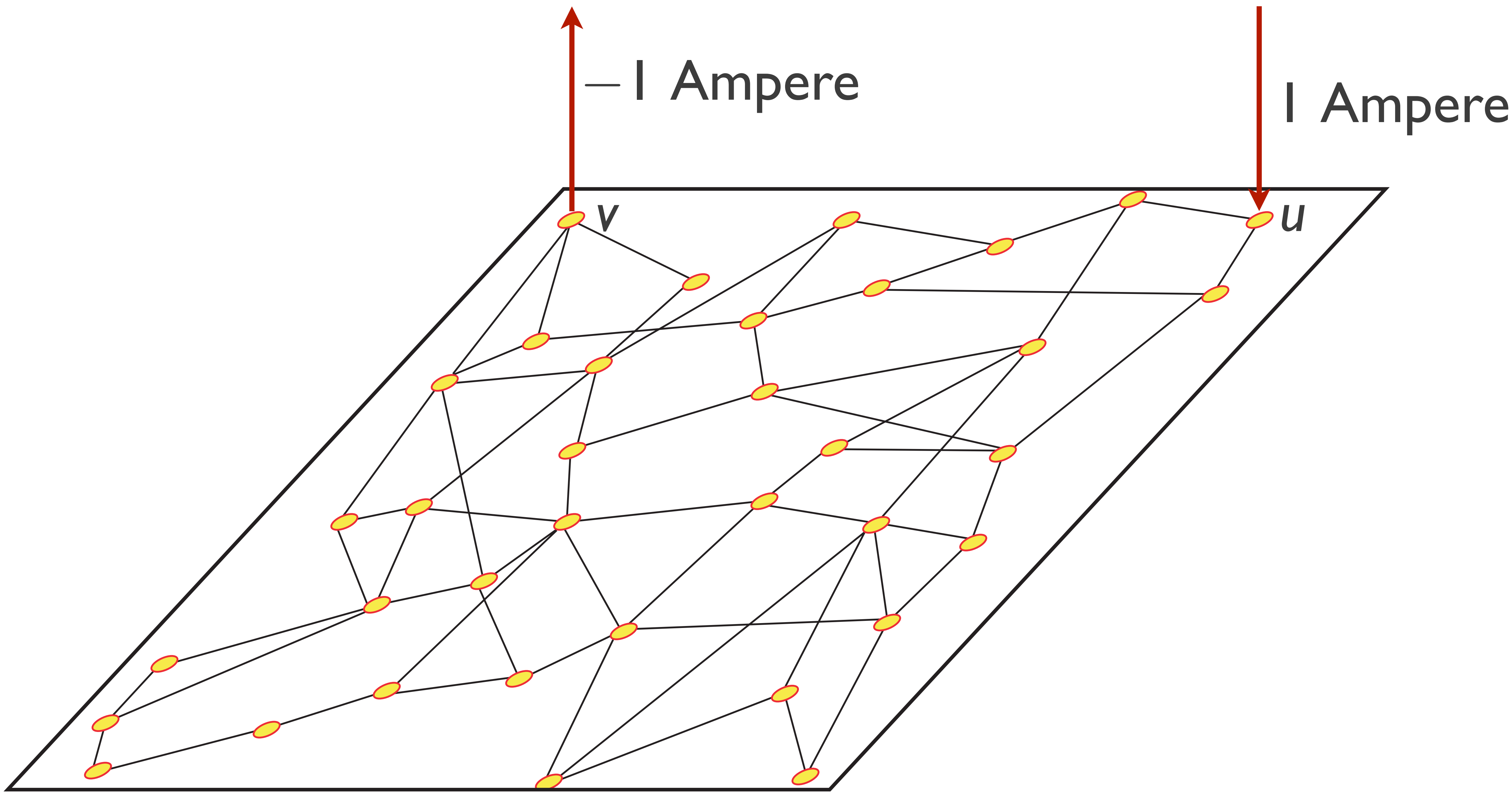
where

$$\boxed{G(Z) := \sum_{t=0}^{\infty} \left(Z^t - \frac{1}{N} \mathbb{1}\mathbb{1}^T \right)}$$ Green function

Therefore

$$v = G(Z) i$$

Electrical network (Doyle, Snell)



Electrical network (Doyle, Snell)

$$J(P) = \frac{1}{N}\operatorname{tr} G(P^2)$$

$$\begin{bmatrix} I - Z & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} G(Z) & N^{-1}\mathbb{1} \\ N^{-1}\mathbb{1}^T & 0 \end{bmatrix}$$

$$\begin{bmatrix} A^T C A & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} i \\ 0 \end{bmatrix} \Longrightarrow \begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} A^T C A & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} i \\ 0 \end{bmatrix}$$

$$\Big\Downarrow \; A^T C A = I - Z$$

$$\boxed{v = G(Z) i} \Longleftarrow \begin{bmatrix} A^T C A & \mathbb{1} \\ \mathbb{1}^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} G(Z) & N^{-1}\mathbb{1} \\ N^{-1}\mathbb{1}^T & 0 \end{bmatrix}$$

Electrical network (Doyle, Snell)

Take $i = e_u - e_v$, where e_u is the column vector with all entries 0 expect for a 1 in position u. Then the effective resistance between u and v is

$$\begin{aligned} \mathcal{R}_{uv} &= v_u - v_v = (e_u - e_v)^T v \\ &= (e_u - e_v)^T G(Z) i \\ &= (e_u - e_v)^T G(Z)(e_u - e_u) \end{aligned}$$

Therefore

$$\boxed{\frac{1}{N^2}\sum_{u \neq v} \mathcal{R}_{uv} = \frac{1}{N} \operatorname{tr} G(Z)}$$

Electrical network (Doyle, Snell)

$$\frac{1}{N^2}\sum_{u\neq v}\mathcal{R}_{uv} = \frac{1}{N}\mathrm{tr}\, G(Z)$$

$$G(Z) := \sum_{t=0}^{\infty}\left(Z^{\,t} - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right)$$

$$J(P) = \frac{1}{N}\mathrm{tr}\sum_{t=0}^{\infty}\left(P^{\,2t} - \frac{1}{N}\right)$$

$$\Rightarrow$$

$$\boxed{J(P) = \frac{1}{N^2}\sum_{u\neq v}\mathcal{R}_{uv}}$$

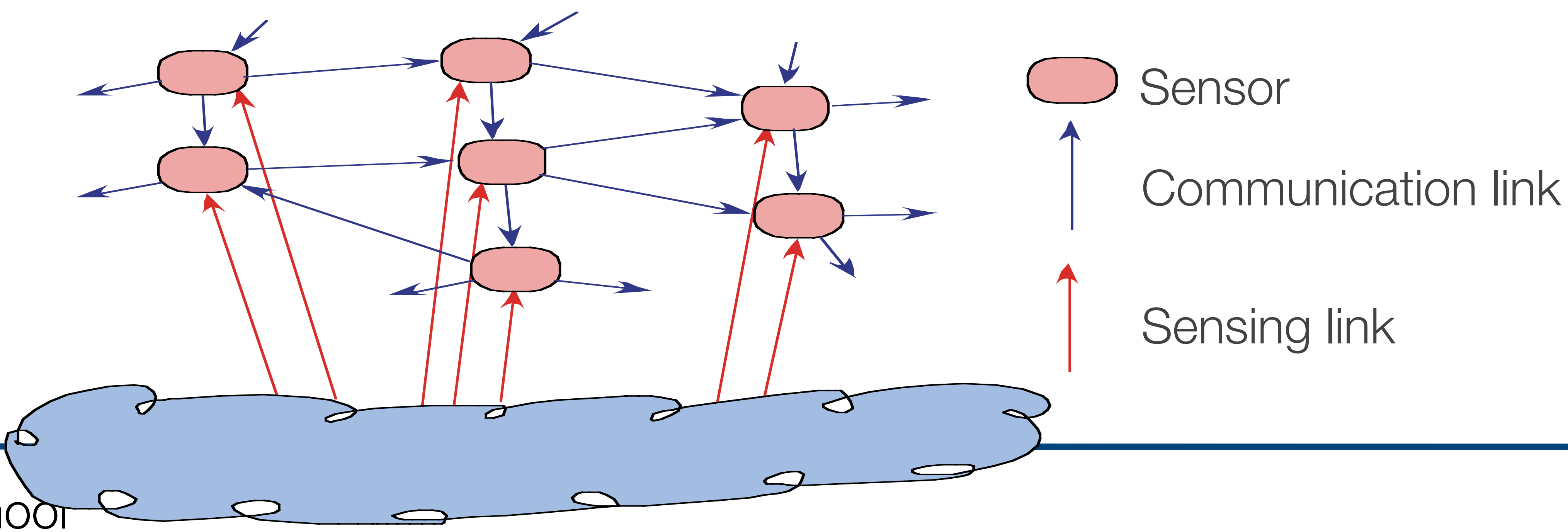
Example: distributed estimation

Assume that N sensors have to estimate a quantity $x \in \mathbb{R}$ from their noisy measurements. The result of the measurement of the sensor i is

$$\boxed{y_i = x + n_i}$$

where n_i are independent noises of zero mean and variance 1. The best estimate of x from the measurements is

$$\boxed{\hat{x} := \frac{1}{N} \sum_i y_i}$$



Example: distributed estimation

SOLUTION: Apply consensus algorithm to find an approximate value of the mean

$$x(t+1) = Px(t)$$
$$x(0) = y$$

$$x_i(t) \simeq z := \frac{1}{N} \sum y_i$$

The natural performance index is the mean square estimation error after t iterations

$$J(P,t) := \frac{1}{N} \sum_i \mathbf{E}[(x_i(t) - z)^2]$$

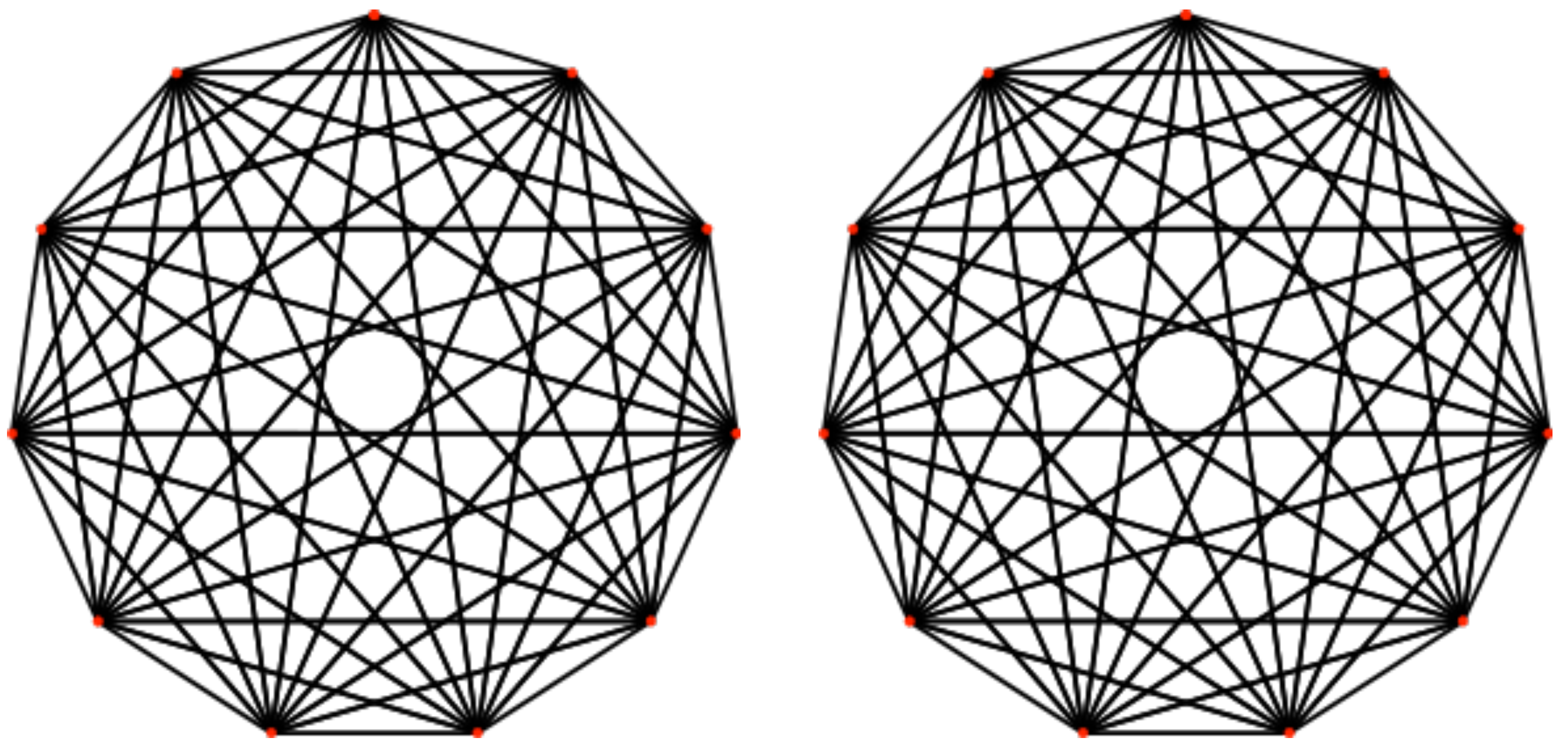
Performance in distributed estimation

It can be shown that (P symmetric)

$$\boxed{J(P,t) = \frac{1}{N}\operatorname{tr} P^{2t} = \frac{1}{N}\sum_{\lambda\in\Lambda(P)} \lambda^{2t}}$$

Performance in distributed estimation

Consider now a graph of N nodes which consists in two disconnected complete graphs with $N/2$ nodes.



In this case

$$\boxed{\rho(P_N) = 1}$$

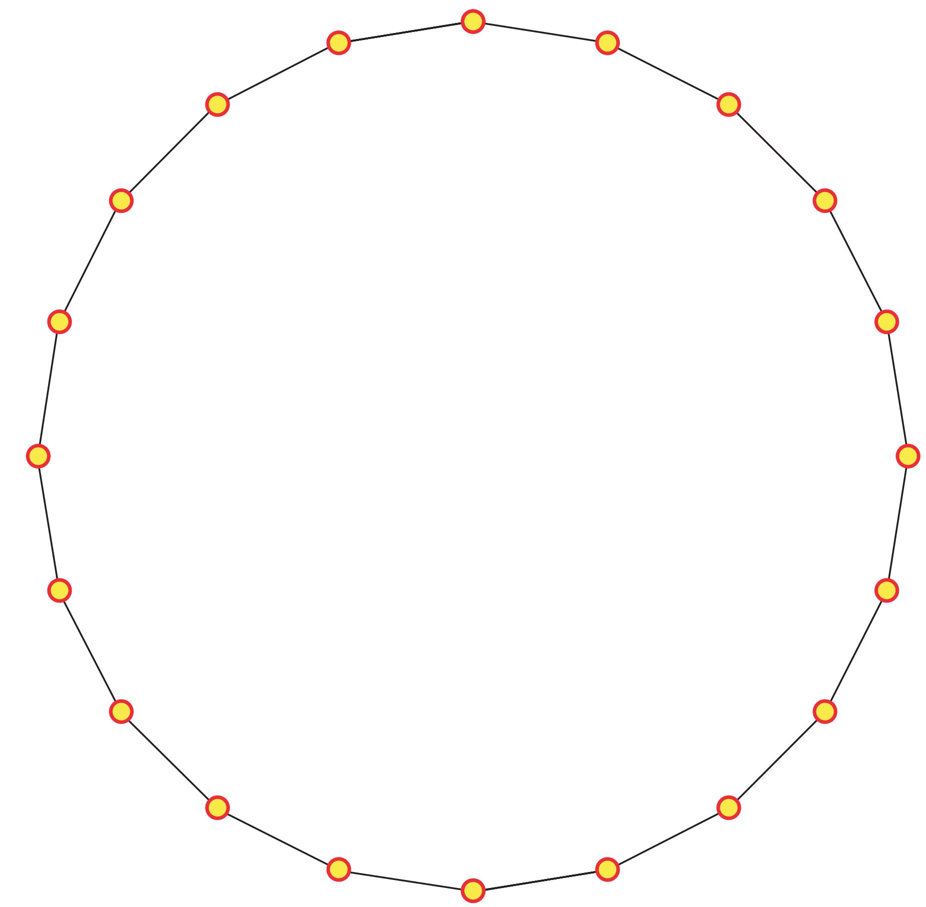
Indeed we will never have consensus in this case. However

$$\boxed{J(P_N, t) = 2/N, \qquad t \geq 1}$$

Performance in distributed estimation

Consider the following $N \times N$ circulant matrix P

$$P_N = \begin{bmatrix} 1/3 & 1/3 & 0 & \cdots & \cdots & 1/3 \\ 1/3 & 1/3 & 1/3 & \cdots & \cdots & 0 \\ 0 & 1/3 & 1/3 & 1/3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1/3 & 1/3 & 1/3 \\ 1/3 & \cdots & \cdots & 0 & 1/3 & 1/3 \end{bmatrix}$$



$$\boxed{\rho(P_N) \simeq 1 - \frac{cost}{N^2} \longrightarrow 1}$$

Performance in distributed estimation

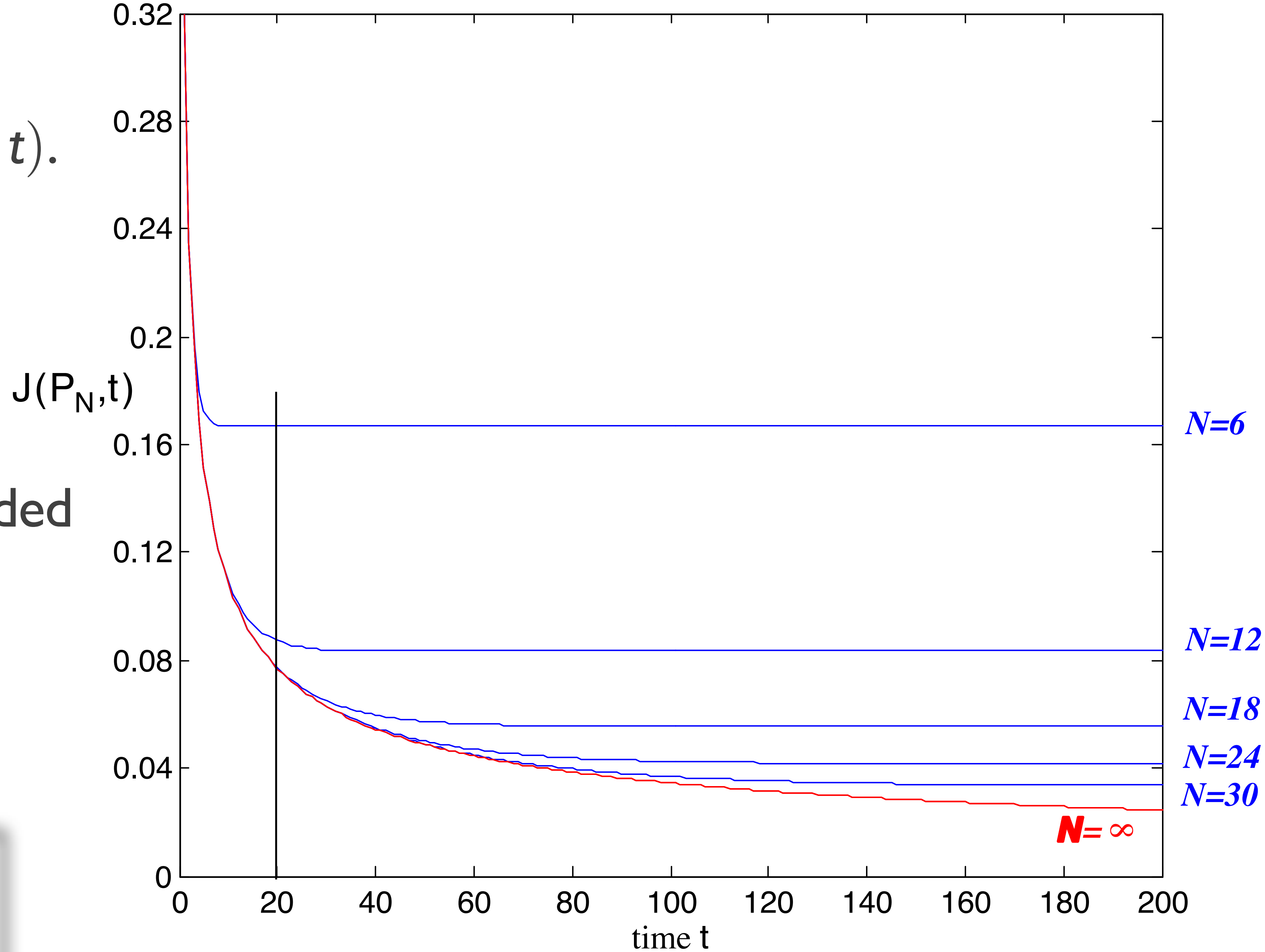
- The bigger is N, the better is the performance $J(P_N, t)$.
- If $t < N/2$ then

$$\boxed{J(P_N, t) = J(P_\infty, t)}$$

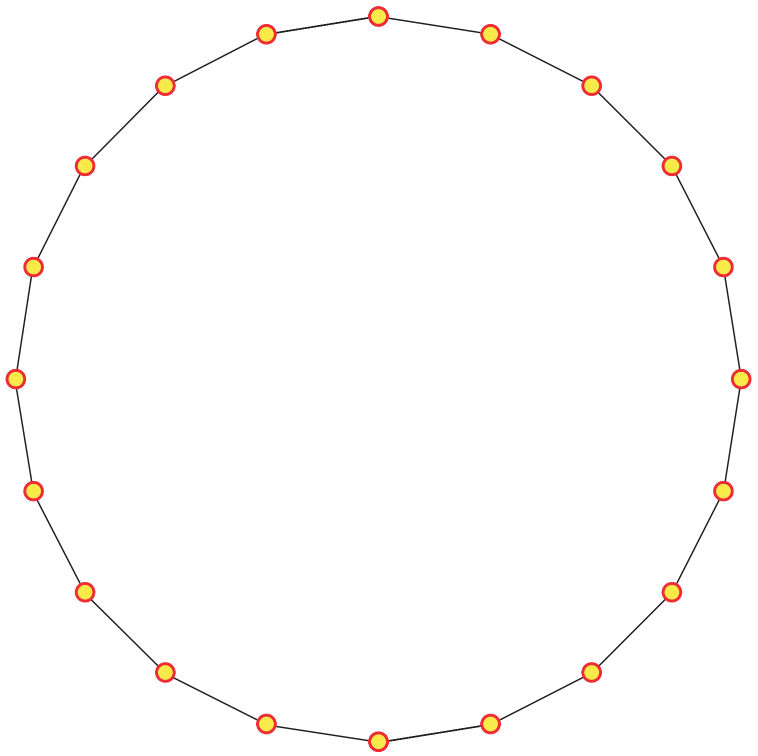
which implies that for bounded t, growing N doesn't buy any gain.

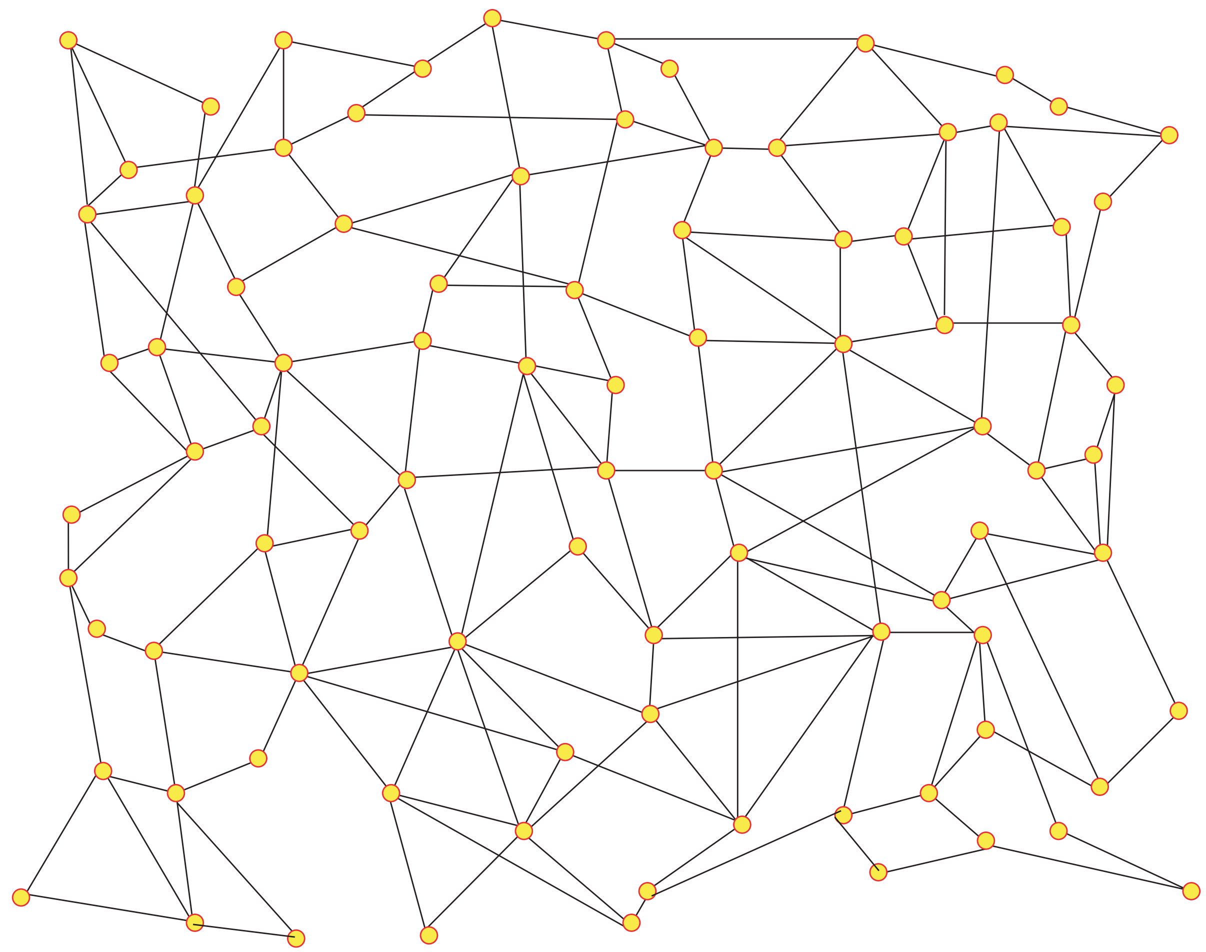
- For N, t big enough

$$\boxed{J(P_N, t) \simeq \max\left\{\frac{1}{N}, \frac{1}{\sqrt{t}}\right\}}$$



Performance in distributed estimation





Can be extended to the torus and the grid and to other geometric graphs with symmetries, but we don't have similar results for more general geometric graphs.

Time varying consensus algorithm

$$x(t+1) = P(t)x(t)$$

$\mathcal{G}_t$ is the graph associated with $P(t)$

Suppose that $P_{ii}(t) > 0$ for all i, t and that there exists a T such that for all ℓ the graph

$$\overline{\mathcal{G}}_\ell := \mathcal{G}_{\ell T} \cup \mathcal{G}_{\ell T+1} \cup \cdots \mathcal{G}_{(\ell-1)T-1}$$

Then

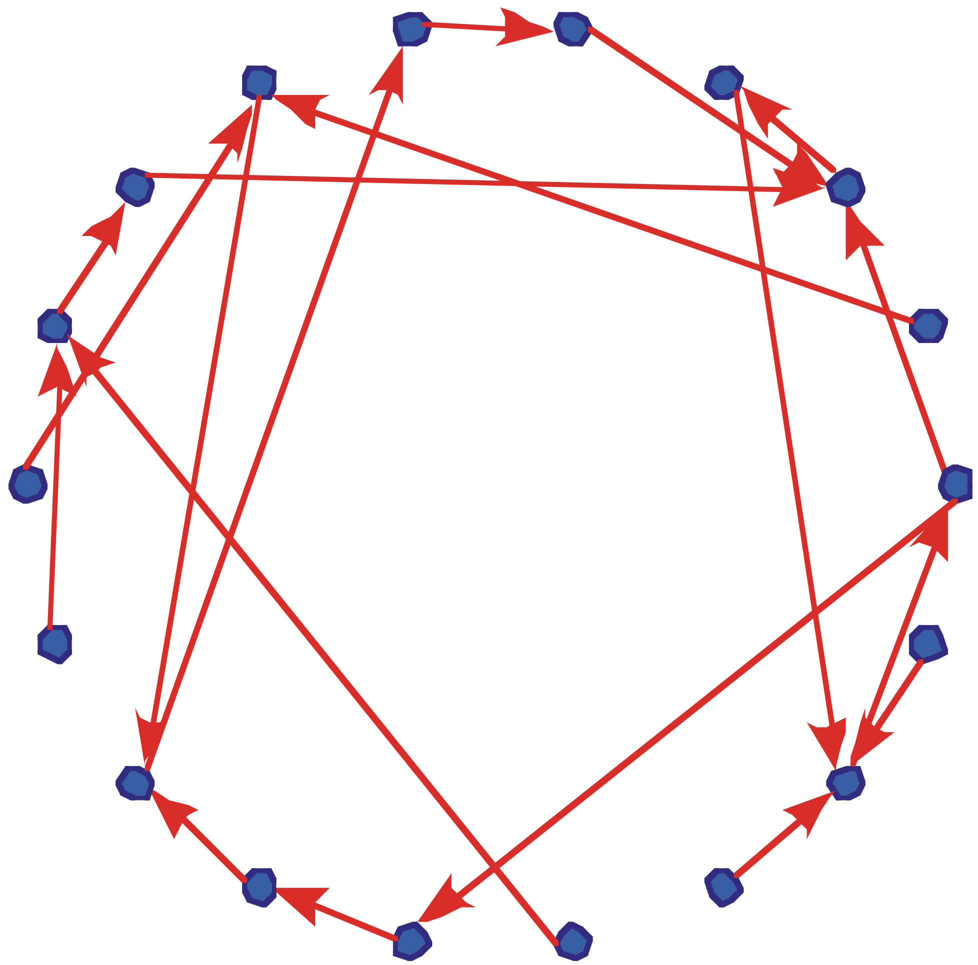
- $x_i(t) \longrightarrow \alpha$ for all i.
- If moreover $P(t)$ are all doubly stochastic, then $x_i(t) \longrightarrow \frac{1}{N}\sum x_i(0)$ for all i.

$\boxed{\text{Estimates of rate of convergence are very conservative (worst case)}}$

Time varying consensus algorithm

$$x(t+1) = P(t)x(t)$$

$\mathcal{G}_t$ is the graph associated with $P(t)$



Suppose that $P_{ii}(t) > 0$ for all i, t and that there exists a T such that for all ℓ the graph

$$\overline{\mathcal{G}}_\ell := \mathcal{G}_{\ell T} \cup \mathcal{G}_{\ell T+1} \cup \cdots \mathcal{G}_{(\ell-1)T-1}$$

Then

- $x_i(t) \longrightarrow \alpha$ for all i.
- If moreover $P(t)$ are all doubly stochastic, then $x_i(t) \longrightarrow \frac{1}{N}\sum x_i(0)$ for all i.

Estimates of rate of convergence are very conservative (worst case)

Performance of randomized consensus algorithms

We now consider randomly time varying stochastic matrices $P(t)$ such that $P(t)_{ii} > 0$ for all i, t. We obtain the system

$$x(t+1) = P(t)x(t)$$

PROBABILISTIC CONSENSUS

$$x_i(t) \to c \qquad \text{almost surely}$$

where $c = \sum \mu_i x_i(0)$ and where μ_i are random variables. Equivalent condition

$$\boxed{P(t-1)\cdots P(0) \to \mathbb{1}\mu^T}$$

where $\mu := (\mu_1, \ldots, \mu_M)^T$ is a random vector.

- Hatano Mesbahi
- Boyd, Ghosh, Prabhakar, Shah
- Tahbaz-Salehi, Jadbabaie
- Porfiri, Stilwell
- Kar, Moura
- Patterson, Bamieh, Abbadi
- Wu
- Fagnani, Zampieri

Performance of randomized consensus algorithms

We now consider randomly time varying stochastic matrices $P(t)$ such that $P(t)_{ii} > 0$ for all i, t. We obtain the system

$$x(t+1) = P(t)x(t)$$

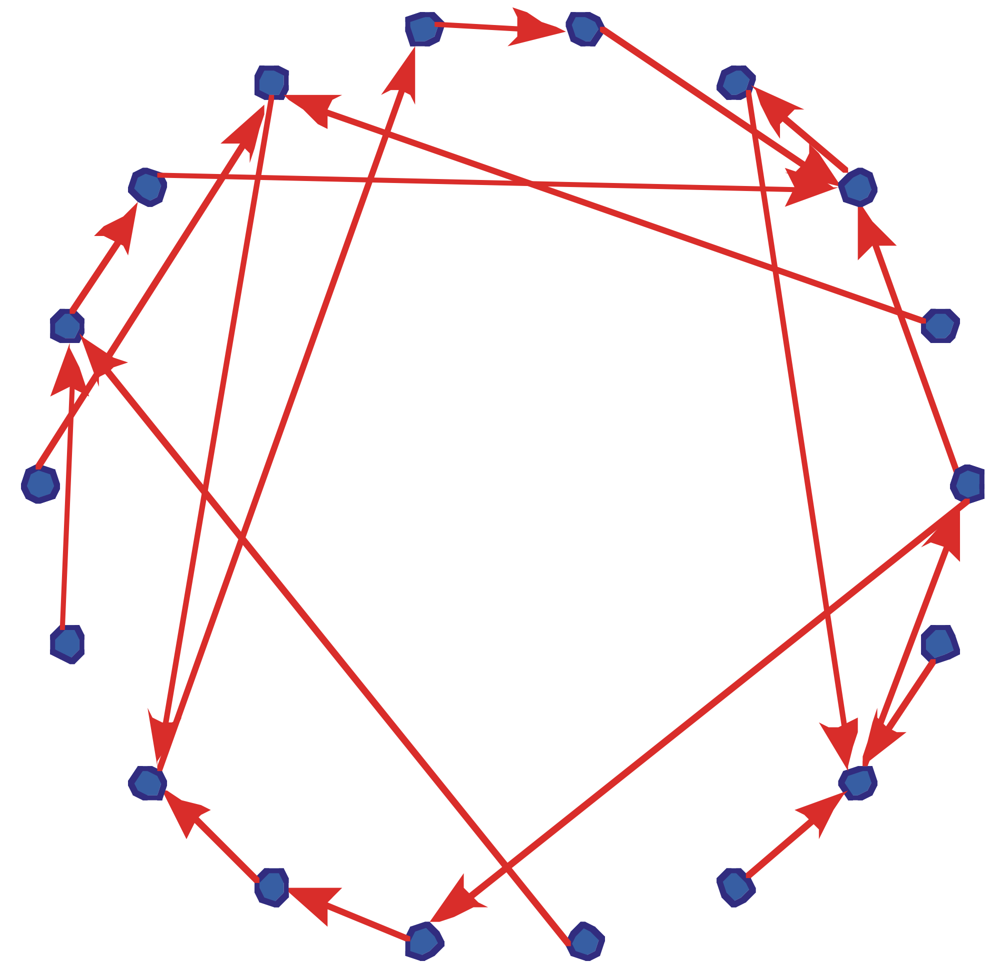
PROBABILISTIC CONSENSUS

$$x_i(t) \to c \qquad \text{almost surely}$$

where $c = \sum \mu_i x_i(0)$ and where μ_i are random variables. Equivalent condition

$$\boxed{P(t-1)\cdots P(0) \to \mathbb{1}\mu^T}$$

where $\mu := (\mu_1, \ldots, \mu_M)^T$ is a random vector.

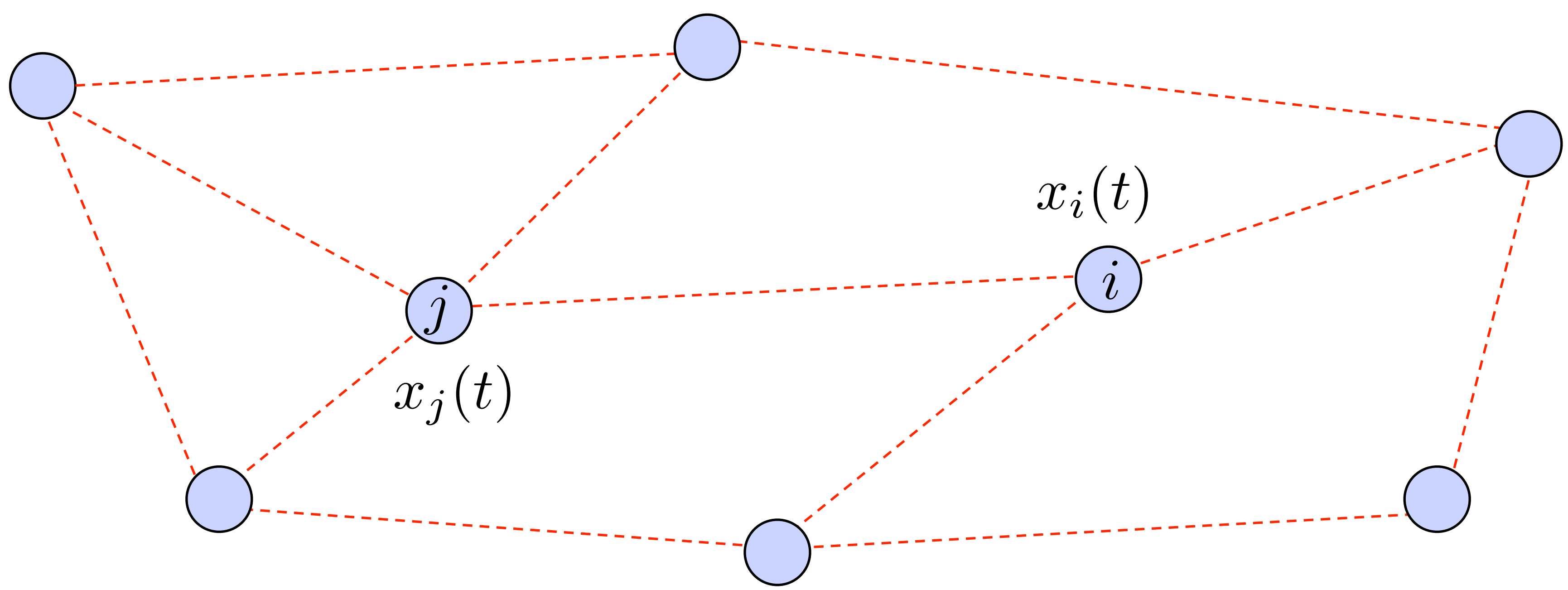


- Hatano Mesbahi
- Boyd, Ghosh, Prabhakar, Shah
- Tahbaz-Salehi, Jadbabaie
- Porfiri, Stilwell
- Kar, Moura
- Patterson, Bamieh, Abbadi
- Wu
- Fagnani, Zampieri

Gossip algorithm (Boyd et al. 2006)

Consider an indirected strongly connected graph $\mathcal{G}$. At each time step, an edge (j, i) is chosen randomly among the edges of $\mathcal{G}$ with probability $p_{ij} > 0$ and the following iteration is done

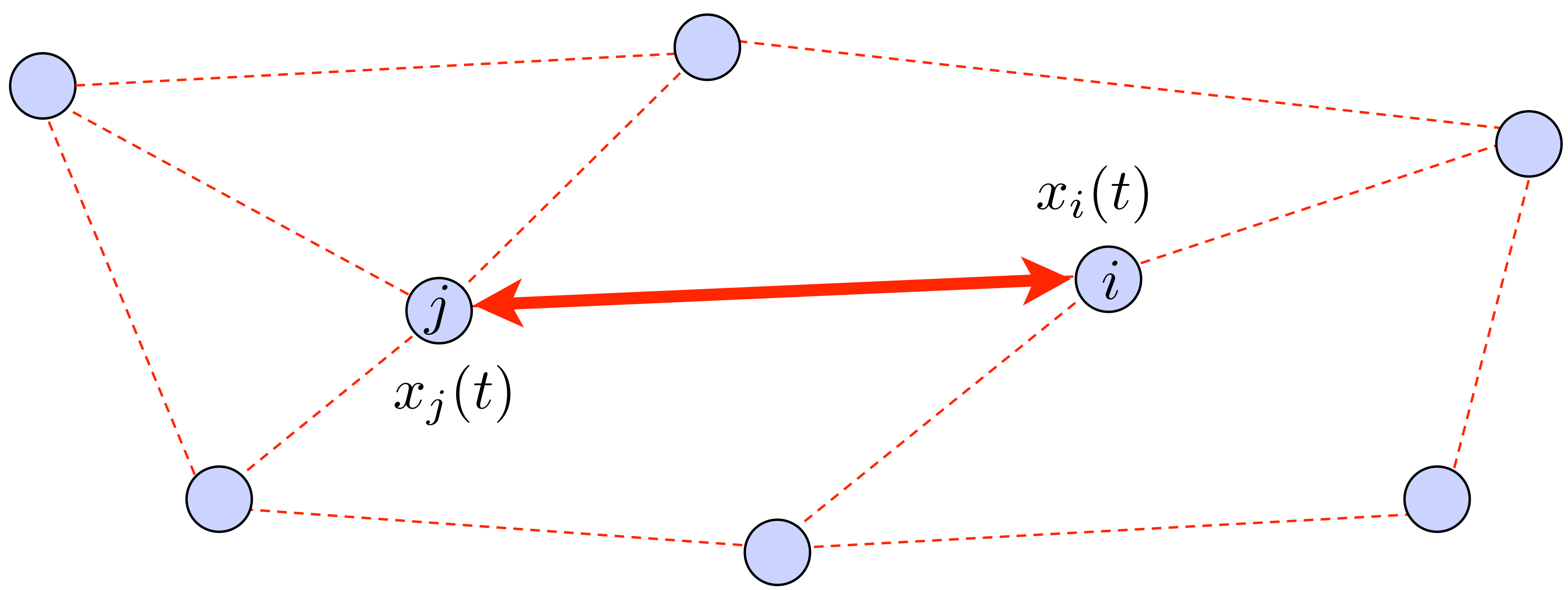
$$\begin{aligned} x_i(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\ x_j(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\ x_h(t+1) &= x_h(t) \qquad h \neq i, j \end{aligned}$$



Gossip algorithm (Boyd et al. 2006)

Consider an indirected strongly connected graph $\mathcal{G}$. At each time step, an edge (j, i) is chosen randomly among the edges of $\mathcal{G}$ with probability $p_{ij} > 0$ and the following iteration is done

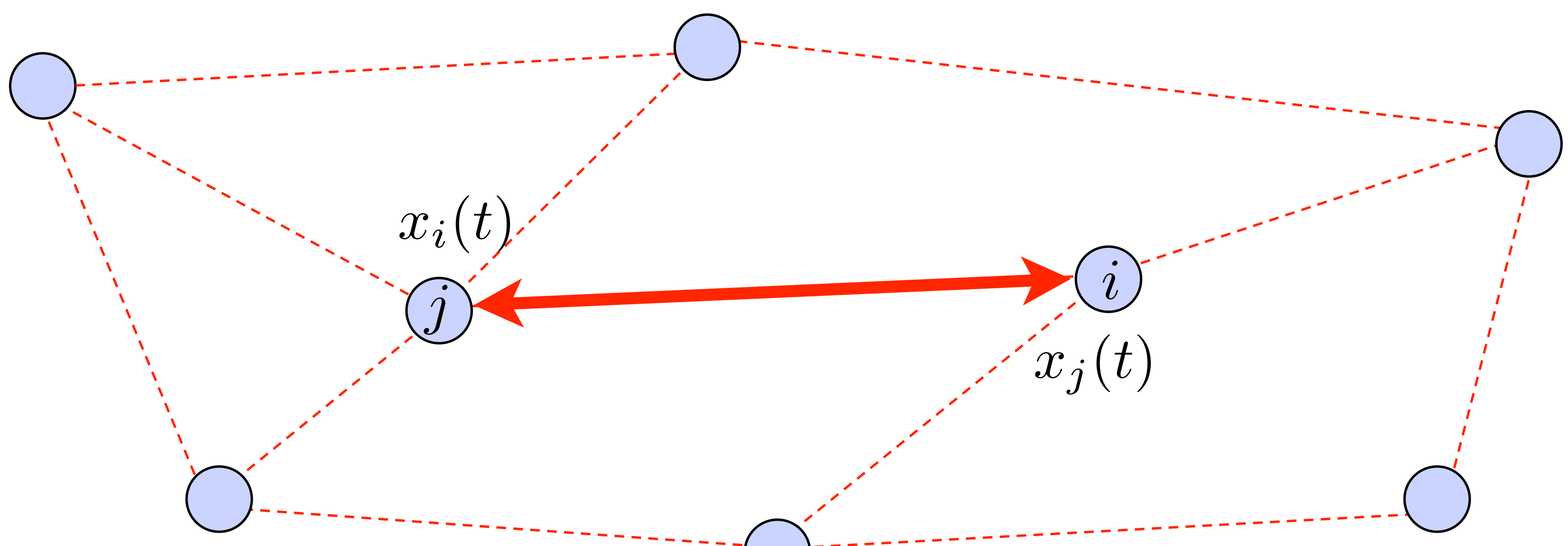
$$\begin{aligned} x_i(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\ x_j(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\ x_h(t+1) &= x_h(t) \qquad h \neq i, j \end{aligned}$$



Gossip algorithm (Boyd et al. 2006)

Consider an indirected strongly connected graph $\mathcal{G}$. At each time step, an edge (j, i) is chosen randomly among the edges of $\mathcal{G}$ with probability $p_{ij} > 0$ and the following iteration is done

$$\begin{aligned} x_i(t+1) &= 1/2 x_i(t) + 1/2 x_j(t) \\ x_j(t+1) &= 1/2 x_i(t) + 1/2 x_j(t) \\ x_h(t+1) &= x_h(t) \qquad h \neq i, j \end{aligned}$$



Gossip algorithm (Boyd et al. 2006)

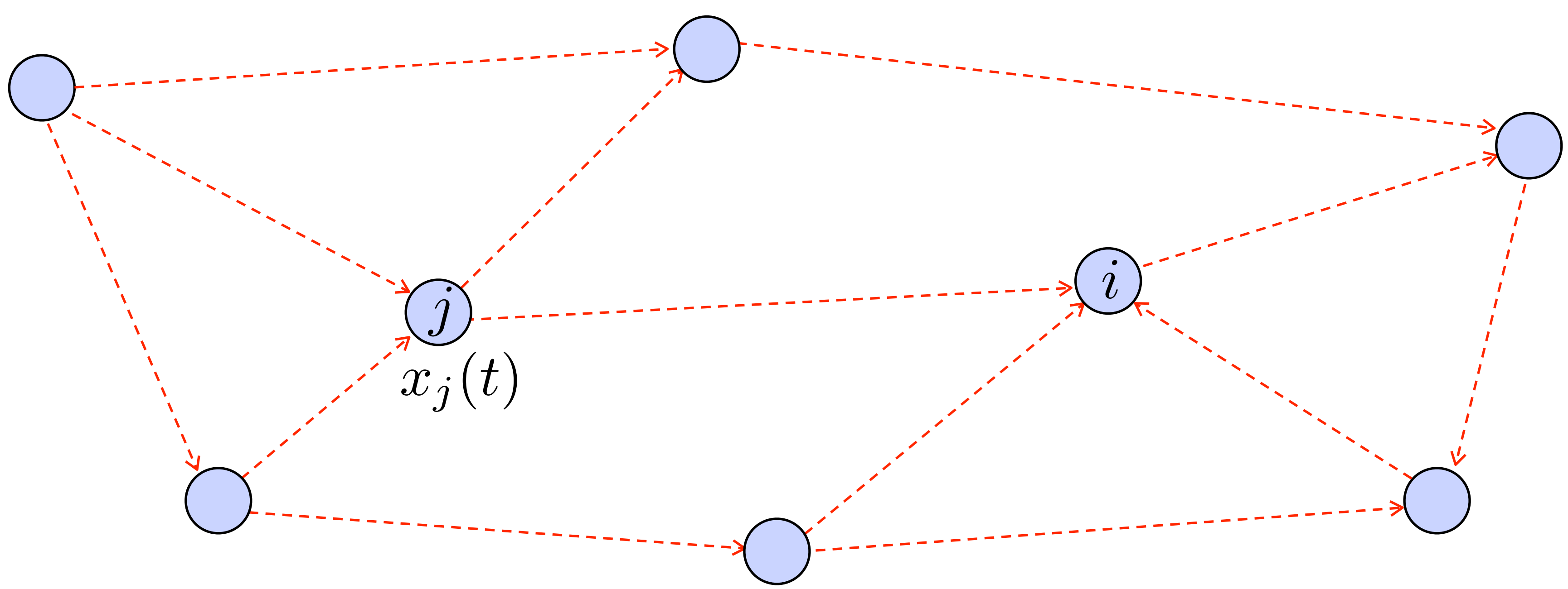
$$
P(t) = \begin{array}{c}
\\ \\ \\ i \\ \\ \\ \\ j \\ \\ \\ \\
\end{array}
\begin{bmatrix}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & & 1/2 & & & & 1/2 & & & \\
 & & & & 1 & & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & & 1 & & & & \\
 & & & 1/2 & & & & 1/2 & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{bmatrix}
$$

$$
\begin{array}{ccccccccccc}
 & & & i & & & & j & & &
\end{array}
$$

Asymmetric gossip algorithm

Consider an directed strongly connected graph $\mathcal{G}$. At each time step, an edge (j, i) is chosen randomly among the edges of $\mathcal{G}$ with probability $p_{ij} > 0$ and the following iteration is done

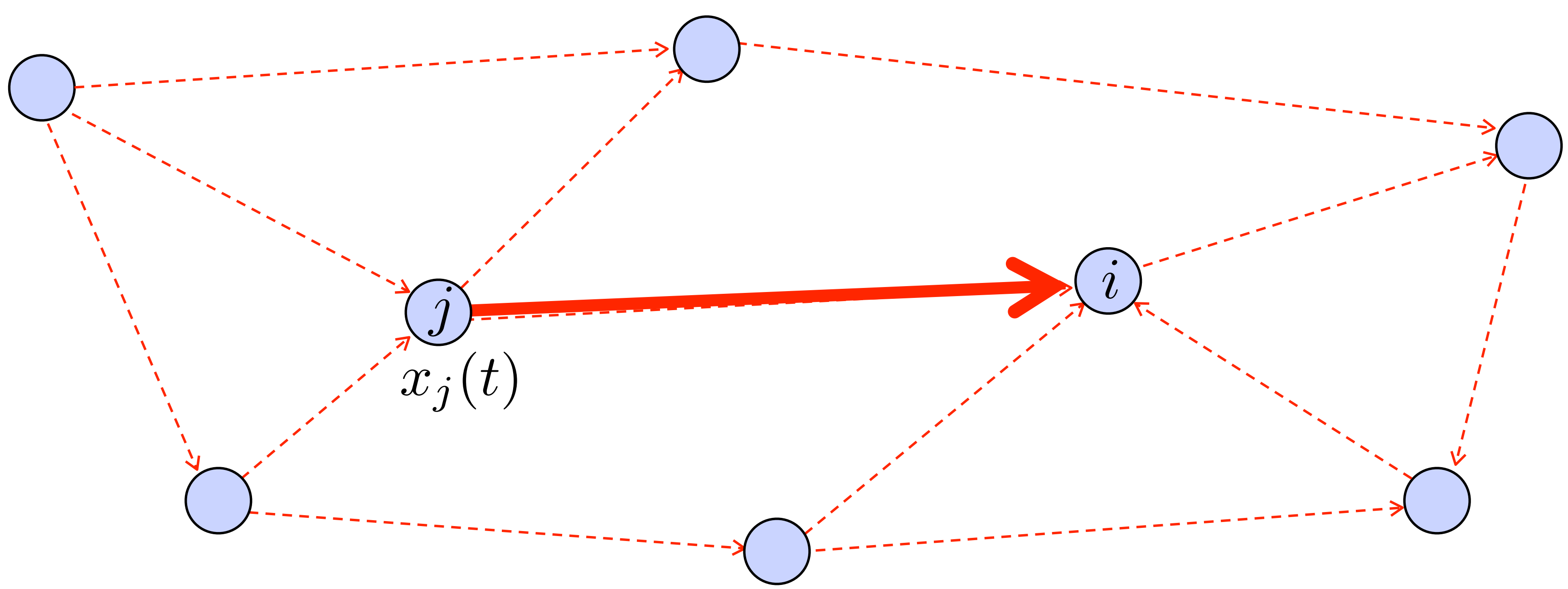
$$\begin{aligned} x_i(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\ x_h(t+1) &= x_h(t) \qquad h \neq i \end{aligned}$$



Asymmetric gossip algorithm

Consider an directed strongly connected graph $\mathcal{G}$. At each time step, an edge (j, i) is chosen randomly among the edges of $\mathcal{G}$ with probability $p_{ij} > 0$ and the following iteration is done

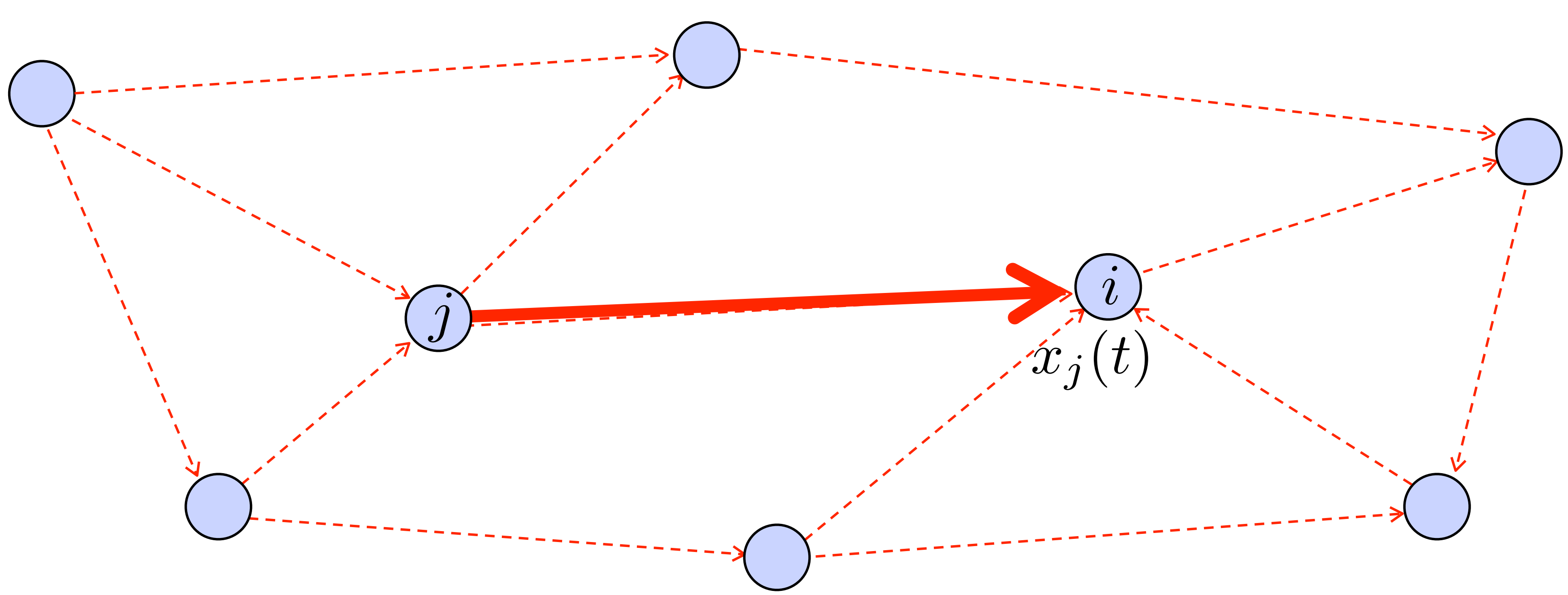
$$
\begin{aligned}
x_i(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\
x_h(t+1) &= x_h(t) \qquad h \neq i
\end{aligned}
$$



Asymmetric gossip algorithm

Consider an directed strongly connected graph $\mathcal{G}$. At each time step, an edge (j, i) is chosen randomly among the edges of $\mathcal{G}$ with probability $p_{ij} > 0$ and the following iteration is done

$$
\begin{aligned}
x_i(t+1) &= 1/2x_i(t) + 1/2x_j(t) \\
x_h(t+1) &= x_h(t) \qquad h \neq i
\end{aligned}
$$

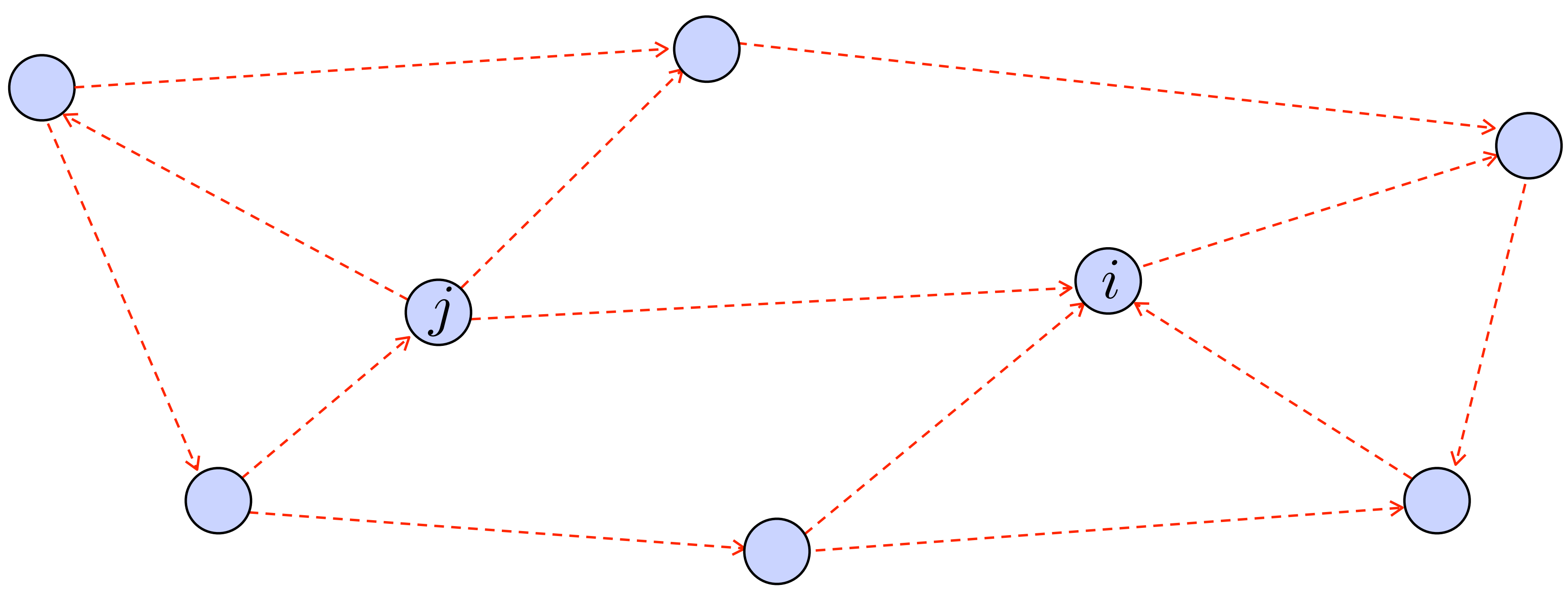


$$
P(t) = \begin{array}{c@{}c}
\begin{array}{c} \\ \\ \\ i \\ \\ \\ \\ \\ \\ \\ \\ \end{array} &
\begin{bmatrix}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & & \boxed{1/2} & & & & \boxed{1/2} & & & \\
 & & & & 1 & & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & & 1 & & & & \\
 & & & & & & & 1 & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{bmatrix} \\
 & \begin{array}{ccccccccccc} & & & & & & & j & & & \end{array}
\end{array}
$$

Broadcast communication

Consider an directed strongly connected graph $\mathcal{G}$. At each time step, a node *i* is chosen randomly and the following iteration is done

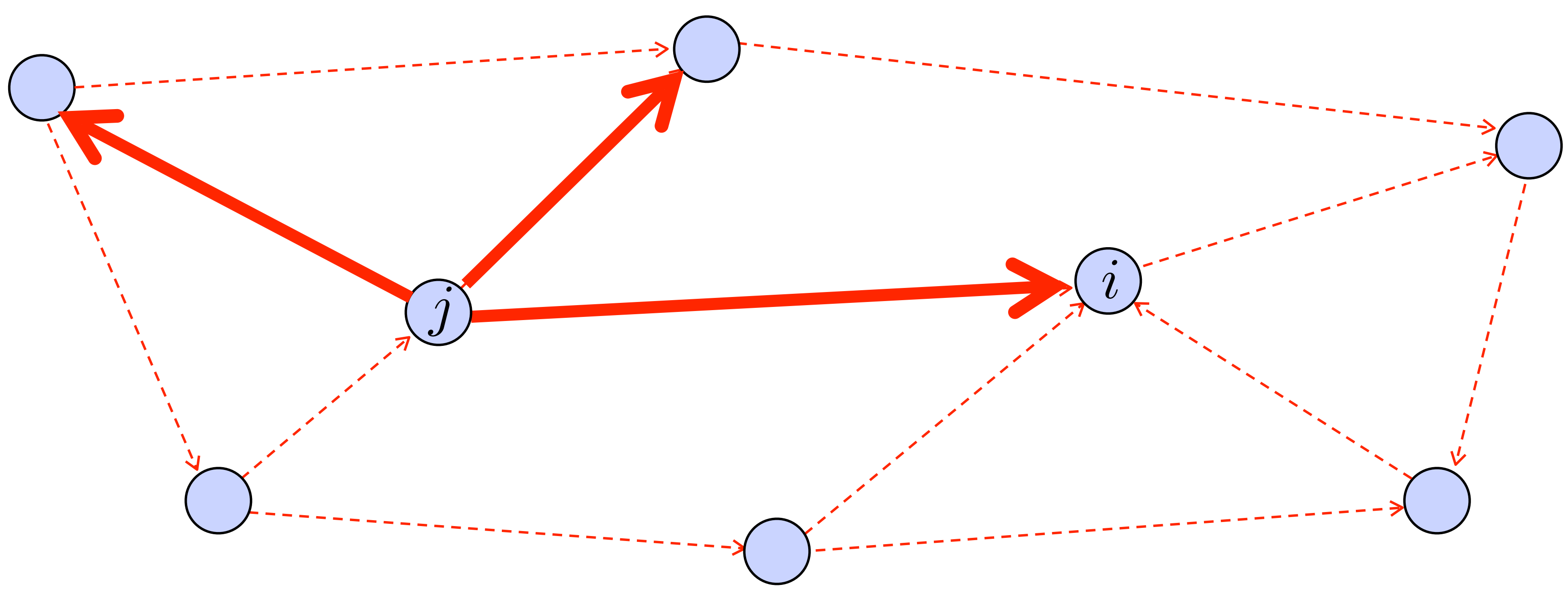
$$
\begin{aligned}
x_j(t+1) &= 1/2x_i(t) + 1/2x_j(t) \qquad \text{for all } j \text{ neighbors of } i \\
x_h(t+1) &= x_h(t) \qquad \text{otherwise}
\end{aligned}
$$



Broadcast communication

Consider an directed strongly connected graph $\mathcal{G}$. At each time step, a node *i* is chosen randomly and the following iteration is done

$$\begin{aligned} x_j(t+1) &= 1/2x_i(t) + 1/2x_j(t) && \text{for all } j \text{ neighbors of } i \\ x_h(t+1) &= x_h(t) && \text{otherwise} \end{aligned}$$



Consensus with packet drops

Patterson, Bamieh, Abbadi
Fagnani, Zampieri
Preciado, Tahbaz-Salehi, Jadbabaie

Assume we start from the standard consensus algorithm

$$x_i(t+1) = \sum_{j=1}^{N} P_{ij} x_j(t)$$

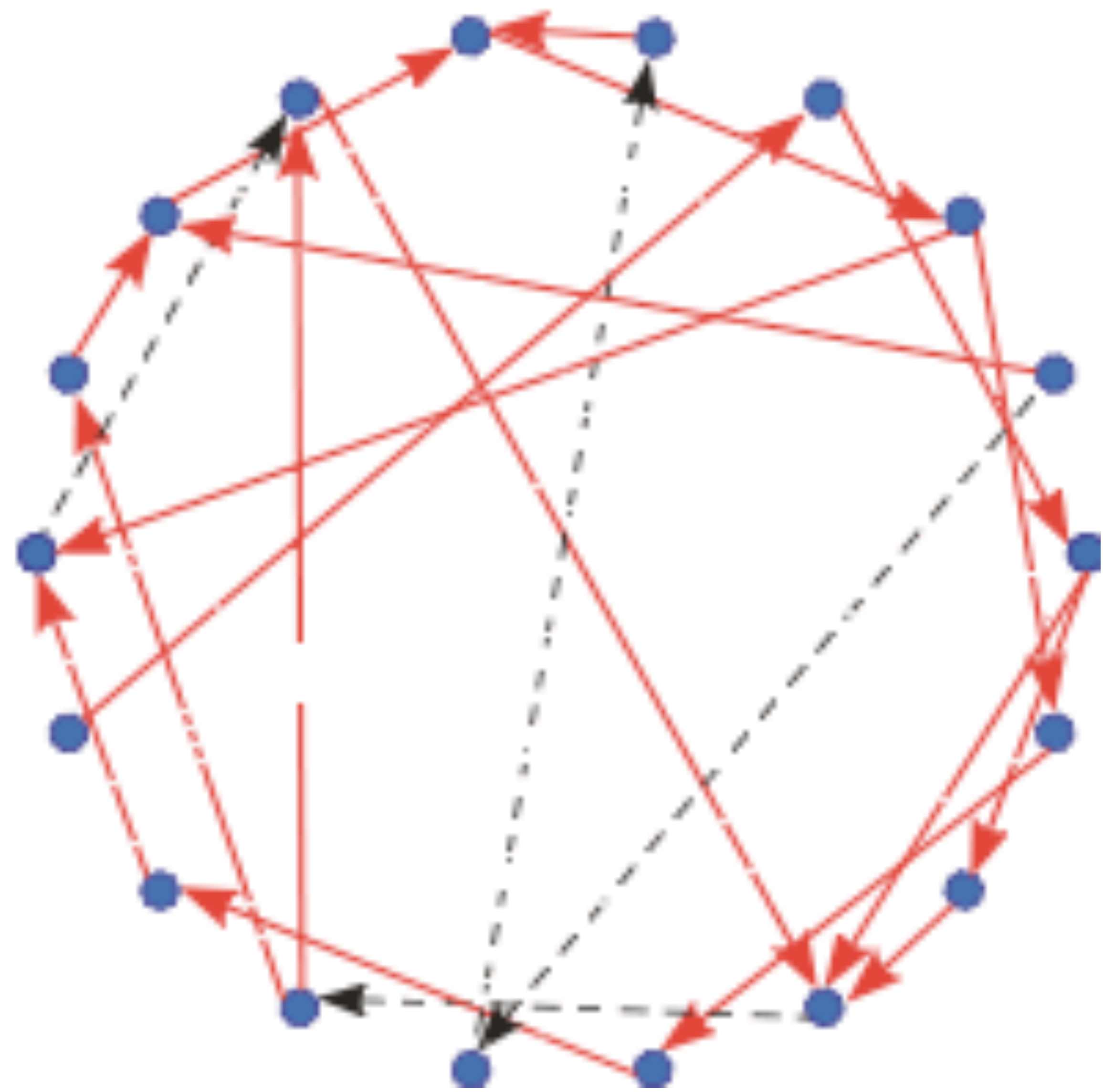
Assume moreover that the binary random variables

$$L_{ij}(t), \quad t \in \mathbf{N}, \quad i,j = 1,\dots,N$$

$i \neq j$, describe the packet loss, namely at time t the packet from j to i is lost iff $L_{ij}(t) = 0$. Assume that those variable are independent and that

$$P[L_{ij}(t) = 1] = p \qquad P[L_{ij}(t) = 0] = 1 - p$$

In this case it is convenient to define, for $i = 1,\dots,N$, the binary random variable $L_{ii}(t)$ which is equal to 1 with probability 1.



Consensus with packet drops

Examples

$$x_i(t+1) = \frac{1}{\sum_{j \in N_i} L_{ij}(t)} \left(\sum_{j \in N_i} L_{ij}(t) x_j(t) \right)$$

or

$$x_i(t+1) = (1 - \epsilon \sum_j L_{ij}) x_i(t) + \epsilon \sum_{j \in N_i \setminus \{i\}} L_{ij}(t) x_j(t)$$

Performance metrics

- Distance from the consensus

$$\boxed{d(t) = \frac{1}{N}||x(t) - \mathbb{1}x_A(t)||^2 = \frac{1}{N}\sum_{i=1}^{N}|x_i(t) - x_A(t)|^2}$$

where $x_A(t) = 1/N \sum x_i(t)$ is the average of the components of $x(t)$.

- Distance from the average of the initial conditions

$$\boxed{\beta(t) = |x_A(t) - x_A(0)|^2}$$

Convergence

THEOREM (Cogburn 1987)

Assume that $P(t)$ are i.i.d. and that $P_{ii}(t) > 0$ almost surely. Then we have probabilistic consensus iff $\bar{P} := \mathbf{E}[P(t)]$ (which is always stochastic) yields deterministic consensus.

Examples

Gossip algorithm

Assume that

$$\boxed{\mathbb{P}[P(t) = I - 1/2(e_i - e_j)(e_i - e_j)^T] = w_{ij}}$$

Then

$$\overline{P}_{ij} = w_{ij}, \qquad \forall i \neq j$$

and so $\mathcal{G}_{\overline{P}} = \mathcal{G}_W$. Moreover

$$P_{ii}(t) \geq 1/2$$

Examples

Packet loss

In this case

$$P_{ii}(t) \geq 1/N$$

Moreover if $i \neq j$ and (j, i) is an edge of the graph, then we have that

$$\begin{aligned}
\bar{P}_{ij} &= \mathbf{E}\left[\frac{L_{ij}(t)}{\sum_{h\in N_i} L_{ih}(t)}\right] \\
&= \mathbf{E}\left[\frac{L_{ij}(t)}{\sum_{h\in N_i} L_{ih}(t)} \,\middle|\, L_{ij}(t) = 0\right] \mathbf{P}[L_{ij}(t) = 0] \\
&+ \mathbf{E}\left[\frac{L_{ij}(t)}{\sum_{h\in N_i} L_{ih}(t)} \,\middle|\, L_{ij}(t) = 1\right] \mathbf{P}[L_{ij}(t) = 1] \\
&= \mathbf{E}\left[\frac{L_{ij}(t)}{\sum_{h\in N_i} L_{ih}(t)} \,\middle|\, L_{ij}(t) = 1\right] p \geq p/|N_i|
\end{aligned}$$

Rate of convergence

THEOREM (Cogburn 1987)

Assume that $P(t)$ are i.i.d. and that $P_{ii}(t) > 0$ almost surely. Then

$$\lim_{t\to\infty} \frac{1}{t} \log ||x(t) - x(\infty)|| = R_{as}^t \quad \text{almost surely}$$

This means that almost surely for big t we have that

$$\boxed{||x(t) - x(\infty)|| \simeq C\, R_{as}^t}$$

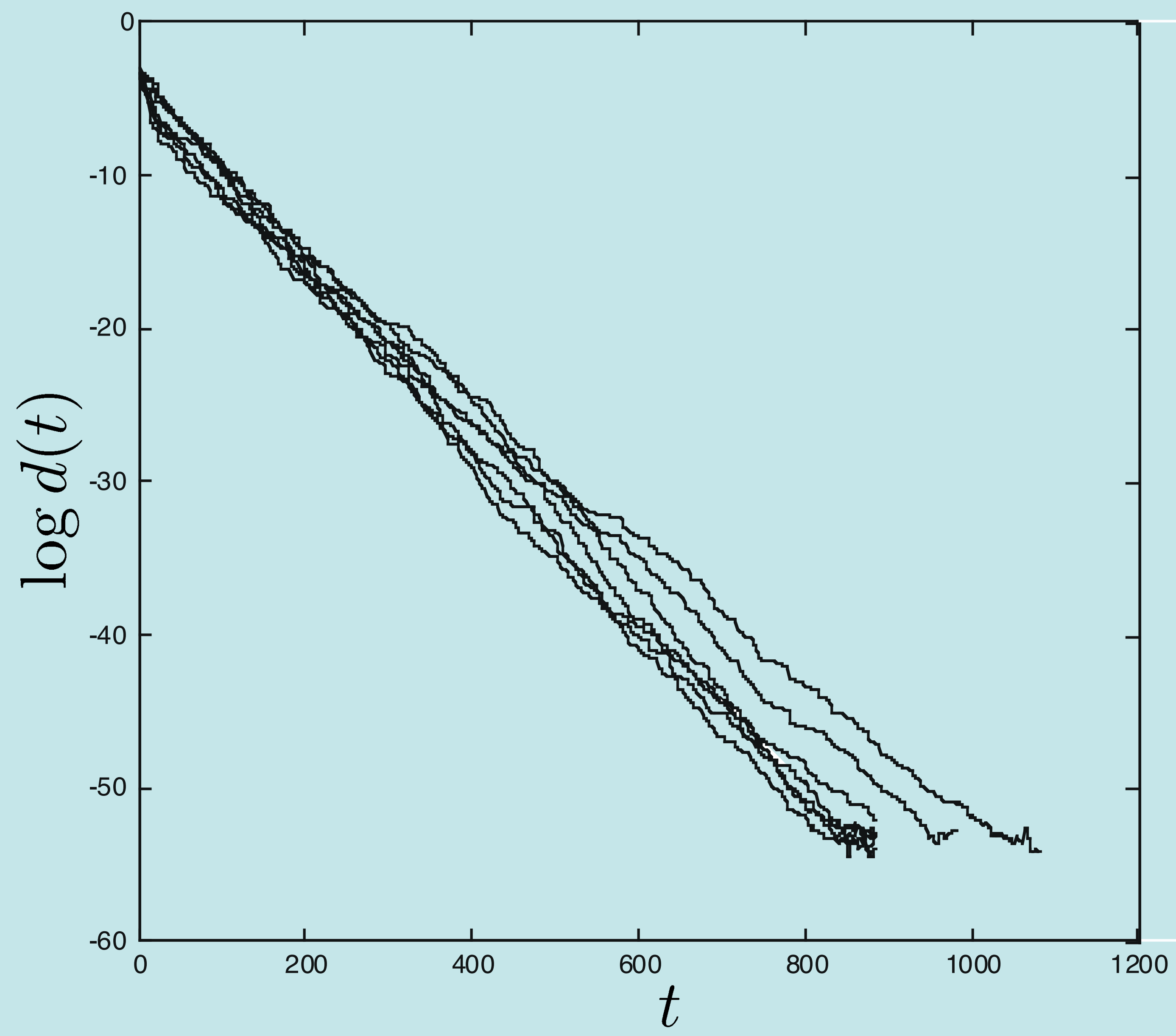
Rate of convergence

EXAMPLE Take the symmetric gossip algorithm with the complete graph and $N = 2^\nu$ nodes and $q = 1/2$. It can be shown easily that the algorithm converges in finite time t_d and so $R_{as} = 0$

$$d(t) = \frac{1}{N}||x(t) - x(\infty)||^2$$

The time t_d is very big, namely $t_d >> N$. Before reaching the consensus we notice from simulations a clear exponential convergence with a certain convergence rate that is different form zero.

The picture shows the logarithmic plot of the trajectories for $N = 8$.



Rate of convergence

Let

$$\bar{d}(t) := \mathbb{E}[d(t)]$$

where

$$d(t) = \frac{1}{N}||x(t) - x(\infty)||^2$$

The rate of convergence of $d(t)$ is different from the rate of convergence of $\bar{d}(t)$.

$$d(t)^{1/2} \simeq CR_{as}^t \text{a.s.} \qquad \bar{d}(t)^{1/2} \simeq CR_{ms}^t \qquad R_{as} \neq R_{ms}$$

THEOREM Assume $P(t)$ yields probabilistic consensus. Then

$$\mathbb{P}[|d(t) - \bar{d}(t)| \geq \delta] \leq \exp\left(-\frac{\delta^2 \alpha(N)}{||x(0)||^4 t}\right)$$

where the function $\alpha(N)$ depends on the probability distribution of $P(t)$.

Typically we have that $\alpha(N) = KN^2$. In this case, if N^2/t is big, then $\mathbb{P}[|d(t) - \bar{d}(t)| \geq \delta]$ is small and so

$$d(t)^{1/2} \simeq \bar{d}(t)^{1/2} \simeq CR_{ms}^t$$

Mean square analysis

Distance from the consensus

Let $\Omega := I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$

$$\mathbf{E}[d(t)] = \frac{1}{N}\mathbf{E}[x^T(t)\Omega x(t)] = \frac{1}{N}x^T(0)\Delta(t)x(0)$$

where

$$\Delta(t) := \mathbf{E}[P(0)^T P(1)^T \cdots P(t-1)^T \Omega P(t-1) \cdots P(1)P(0)]$$

where $\Delta(0) := \Omega$. The dynamics is described by

$$\boxed{\Delta(t+1) = \mathcal{L}(\Delta(t))}$$

The linear operator $\mathcal{L} : \mathbf{R}^{N\times N} \to \mathbf{R}^{N\times N}$ is given by

$$\mathcal{L}(M) = \mathbf{E}[P(0)^T M P(0)]$$

Mean square analysis

Distance from the consensus

Linear dynamic system

$$\boxed{\begin{cases} \Delta(t+1) = \mathcal{L}(\Delta(t)) & \Delta(0) := \Omega \\ \mathbf{E}[d(t)] = \frac{1}{N} x^T(0)\Delta(t)x(0) & \end{cases}}$$

$$\begin{cases} \delta(t+1) = \mathbf{L}\delta(t) \\ \mathbf{E}[d(t)] = C(x(0))\delta(t) \end{cases}$$

where $\mathbf{L} := \mathbf{E}[P^T(0) \otimes P^T(0)]$. We have that

$$\delta(t) = \mathrm{vect}(\Delta(t))$$

Notice that $\mathbf{L}^T$ is a stochastic matrix.

Mean square analysis

Distance from the consensus

We know that $\mathbf{E}[d(t)]$ converges to zero exponentially. Let

$$R_{ms} := \lim_{t\to\infty} \frac{1}{t} \log \mathbf{E}[d(t)]$$

Let Sym the be the set of $N \times N$ complex hermitian matrices.

PROPOSITION

- $\mathcal{L}_{|\mathrm{Sym}}$ has only real eigenvalues.
- R_{ms} is the second largest absolute value of the eigenvalues of $\mathcal{L}_{|\mathrm{Sym}}$.

PROPOSITION

- $\boxed{\mathrm{esr}\left(\overline{P}\right)^2 \leq R_{ms} \leq \mathrm{sr}(\mathcal{L}(\Omega))}$ where $\mathrm{esr}(\cdot)$ means second largest eigenvalue and $\mathrm{sr}(\cdot)$ means the largest eigenvalue.

Examples

Complete graph

- Gossip with probability $p_{ij} = 1/N^2$

$$R_{ms}^2 = 1 - \frac{1}{N}$$

- Packet drop

$$R_{ms}^2 = (1 - \beta N)(1 - \beta)$$

where $\beta := p\mathbb{E}\left[\frac{1}{2+b_{N-2}}\right]$ and b_k is the binomial random variable namely a random variable taking value on $\{0, 1, \ldots, k\}$ with law $\mathbb{P}[b_k = i] = \binom{k}{i} p^i (1-p)^{k-i} \quad i = 0, 1, \ldots, k.$

Distance of the consensus value from the average

Observe that $\beta(\infty) = |x_A(\infty) - x_A(0)|^2 = |(\mu^T - N^{-1}\mathbf{1}^T)x(0)|^2$, and so

$$\boxed{\mathbf{E}[\beta(\infty)] = x(0)^T B x(0)}$$

where, if $\mathbb{E}[P(t)]$ is doubly stochastic,

$$B = \mathbf{E}[\mu\mu^T] - N^{-2}\mathbf{1}\mathbf{1}^T$$

Notice moreover that

$$W := \mathbf{E}[\mu\mu^T] = \frac{1}{N}\lim_{t\to\infty}\mathcal{L}^t(I)$$

and so W is the only matrix such that $\mathcal{L}(W) = W$ and $\mathbf{1}^T W \mathbf{1} = 1$

Distance of the consensus value from the average

Observe that $\beta(\infty) = |x_A(\infty) - x_A(0)|^2 = |(\mu^T - N^{-1}\mathbf{1}^T)x(0)|^2$, and so

$$\boxed{\mathbf{E}[\beta(\infty)] = x(0)^T B x(0)}$$

where

$$B = \mathbf{E}[\mu\mu^T] - \frac{1}{N}\mathbf{E}[\mu]\mathbf{1}^T - \frac{1}{N}\mathbf{1}\mathbf{E}[\mu]^T + N^{-2}\mathbf{1}\mathbf{1}^T$$

When $\overline{P}$ is doubly stochastic, we obtain in particular

$$B = \mathbf{E}[\mu\mu^T] - N^{-2}\mathbf{1}\mathbf{1}^T$$

Notice moreover that

$$W := \mathbf{E}[\mu\mu^T] = \frac{1}{N}\lim_{t\to\infty}\mathcal{L}^t(I)$$

and so W is the only matrix such that $\mathcal{L}(W) = W$ and $\mathbf{1}^T W \mathbf{1} = 1$

Examples

EXAMPLE: ASYMMETRIC GOSSIP In this case, under the condition

$$\sum_{j=1}^{N} p_{ij} = \sum_{j=1}^{N} p_{ji}$$

the eigenvector of the operator $\mathcal{L}$ can be computed explicitly and so we obtain that

$$B = \frac{1}{N(N+1)}\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$$

Notice that B converges to zero as N tends to infinity.

EXAMPLE: PACKET LOSS WITH COMPLETE GRAPH: ERDOS RENIY GRAPH

It can be shown in this case that

$$B = \frac{1}{N^2}\frac{1-N\beta}{1-\beta+1/N}\left(I-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$$

where $\beta := p\mathbb{E}\left[\frac{1}{2+b_{N-2}}\right]$ and b_k is the binomial random variable namely a random variable taking value on $\{0,1,\dots,k\}$ with law $\mathbb{P}[b_k = i] = \binom{k}{i} p^i(1-p)^{k-i} \quad i = 0,1,\dots,k$. Moreover, since $\frac{1-N\beta}{1-\beta+1/N} \leq 1$, we have that

$$\boxed{B \leq \frac{1}{N^2}\left(I-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)}$$

and so it tends to zero as N^{-2} as N tends to ∞.

Higher order consensus

Problem Consider the following consensus algorithm

$$
\begin{aligned}
x_i(h+1) &= Ax_i(h) + Bu_i(h) \\
y_i(h) &= Cx_i(h) \\
u_i(h) &= \sum_{j=1}^{N} K_{ij}(h)\, F(x_j(h) - x_i(h))
\end{aligned}
$$

Higher order consensus

Problem Consider the following consensus algorithm

$$
\begin{aligned}
x_i(h+1) &= Ax_i(h) + Bu_i(h) \\
y_i(h) &= Cx_i(h) \\
u_i(h) &= \sum_{j=1}^{N} K_{ij}(h)\, F(x_j(h) - x_i(h)) \boxed{+Ex_i(h)}
\end{aligned}
$$

Higher order consensus

Problem Consider the following consensus algorithm

$$\begin{aligned} x_i(h+1) &= Ax_i(h) + Bu_i(h) \\ y_i(h) &= Cx_i(h) \\ u_i(h) &= \sum_{j=1}^{N} K_{ij}(h)\, F(x_j(h) - x_i(h)) \boxed{+Ex_i(h)} \end{aligned}$$

Determine conditions on the consensus matrix $K(h)$ and on F, E ensuring that

$$|y_i(h) - y_j(h)| \to 0 \qquad \forall i, j$$

Higher order consensus: another example

It has been shown (Muthukrishnan, Ghosh, Schultz, 1998) that second order consensus

$$\boxed{x(h+1) = \beta P x(h) + (1-\beta)x(h-1)}$$

may yield to faster convergence rate with a suitable choice of β. Simulations support that the same holds for consensus with (randomly) time varying graph topology

$$x(h+1) = \beta P(h) x(h) + (1-\beta)x(h-1)$$

Higher order consensus: another example

This can be seen as higher order consensus

$$
\begin{aligned}
x_i(h+1) &= Ax_i(h) + Bu_i(h) \\
y_i(h) &= Cx_i(h) \\
u_i(h) &= \sum_{j=1}^{N} K_{ij}(h)\, F(x_j(h) - x_i(h)) + Ex_i(h)
\end{aligned}
$$

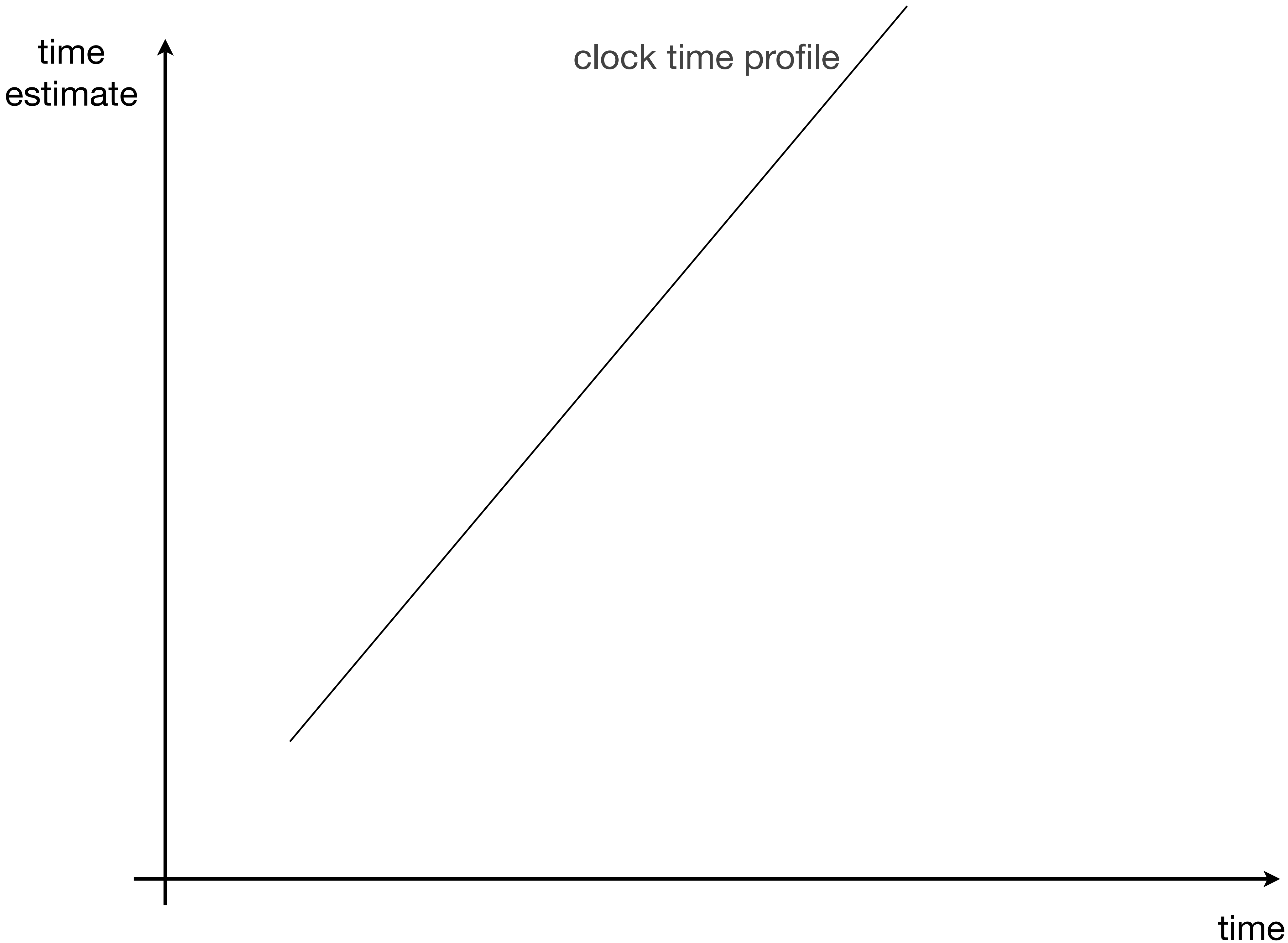
with $K(h) := I - P(h)$ and

$$
A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}
$$

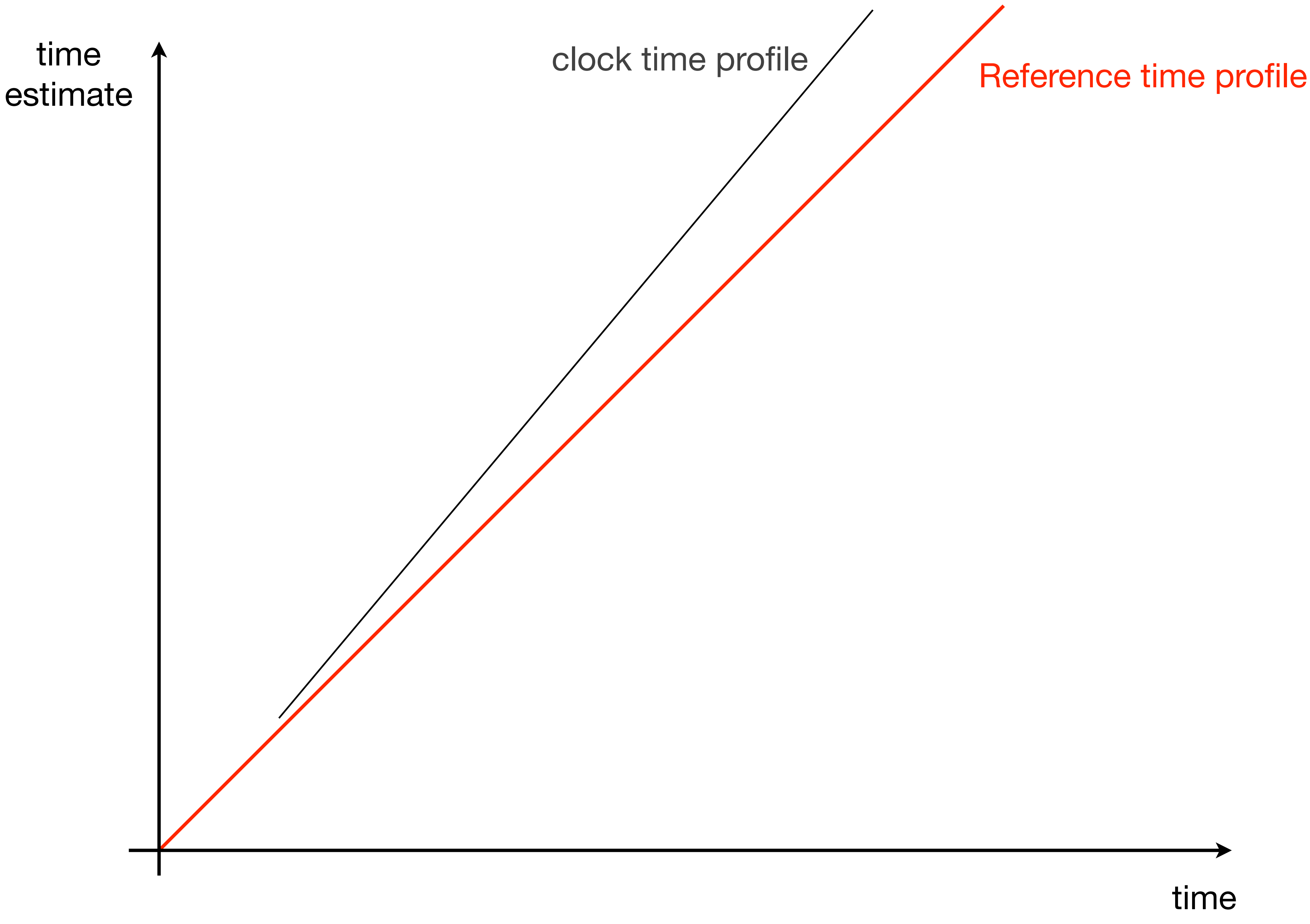
$$
F = \begin{bmatrix} -\beta & 0 \\ 0 & 0 \end{bmatrix}, \qquad E = \begin{bmatrix} 1-\beta & \beta - 1 \\ 0 & 0 \end{bmatrix}
$$

Clock synchronization

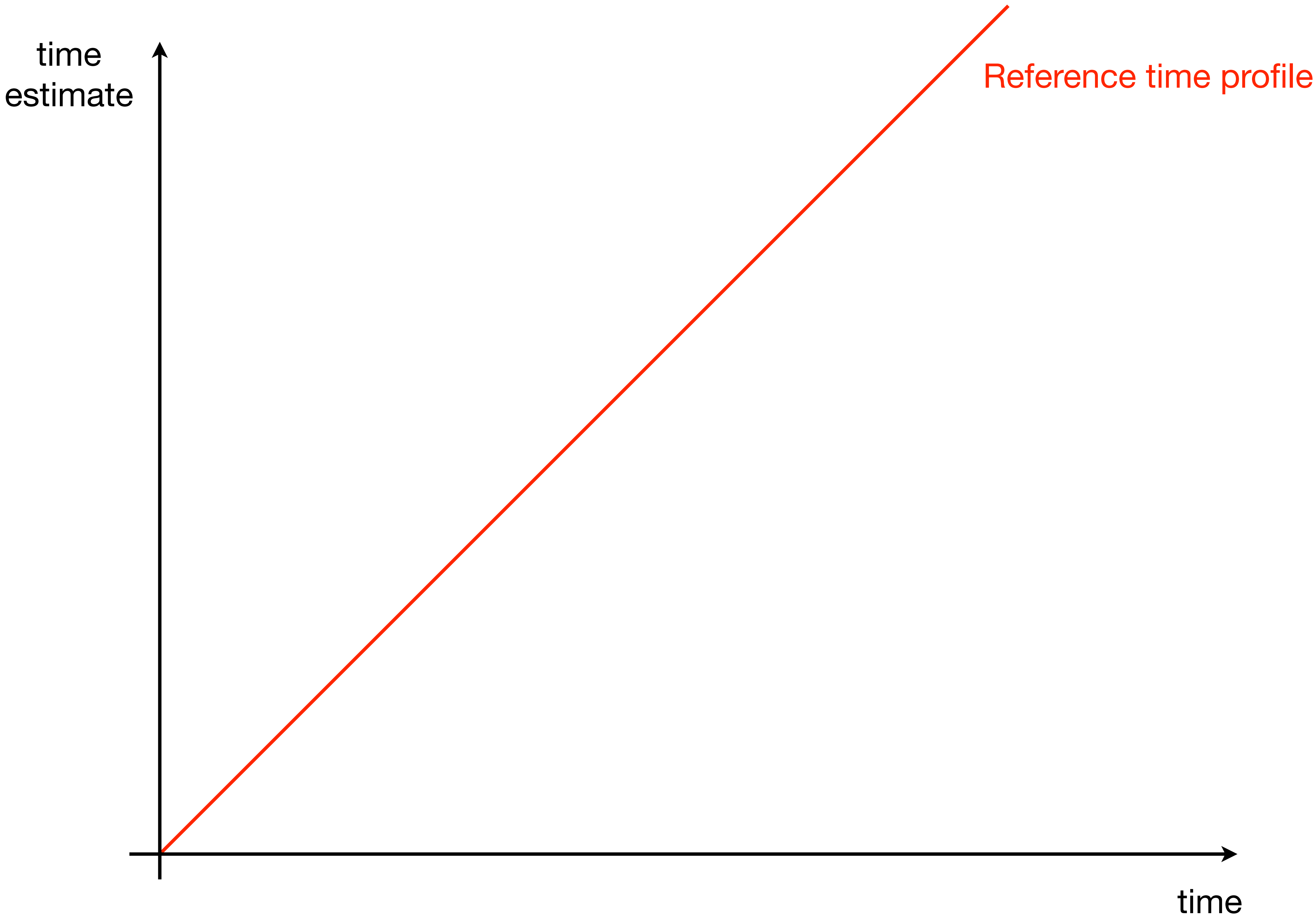
Clock synchronization



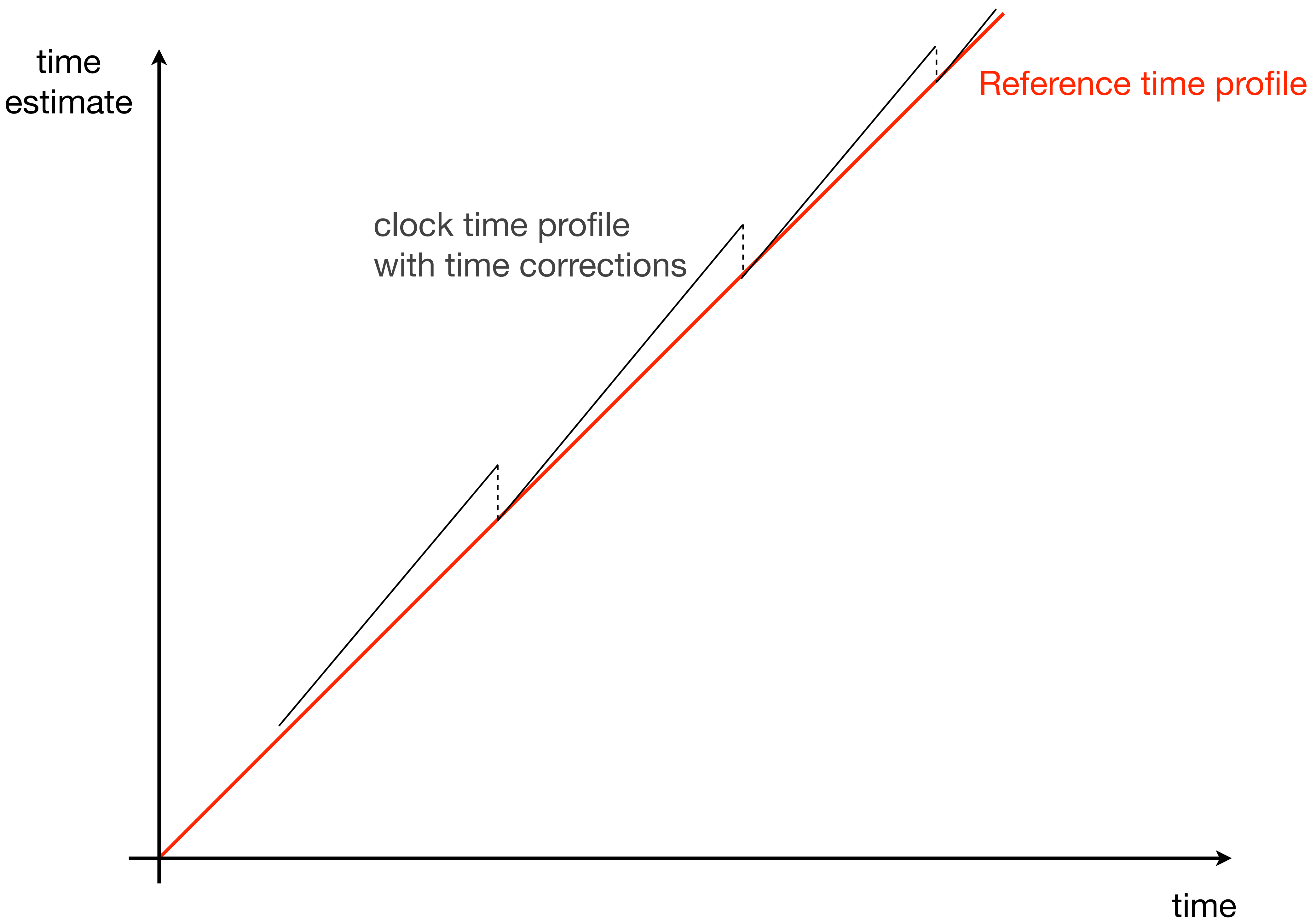
Clock synchronization



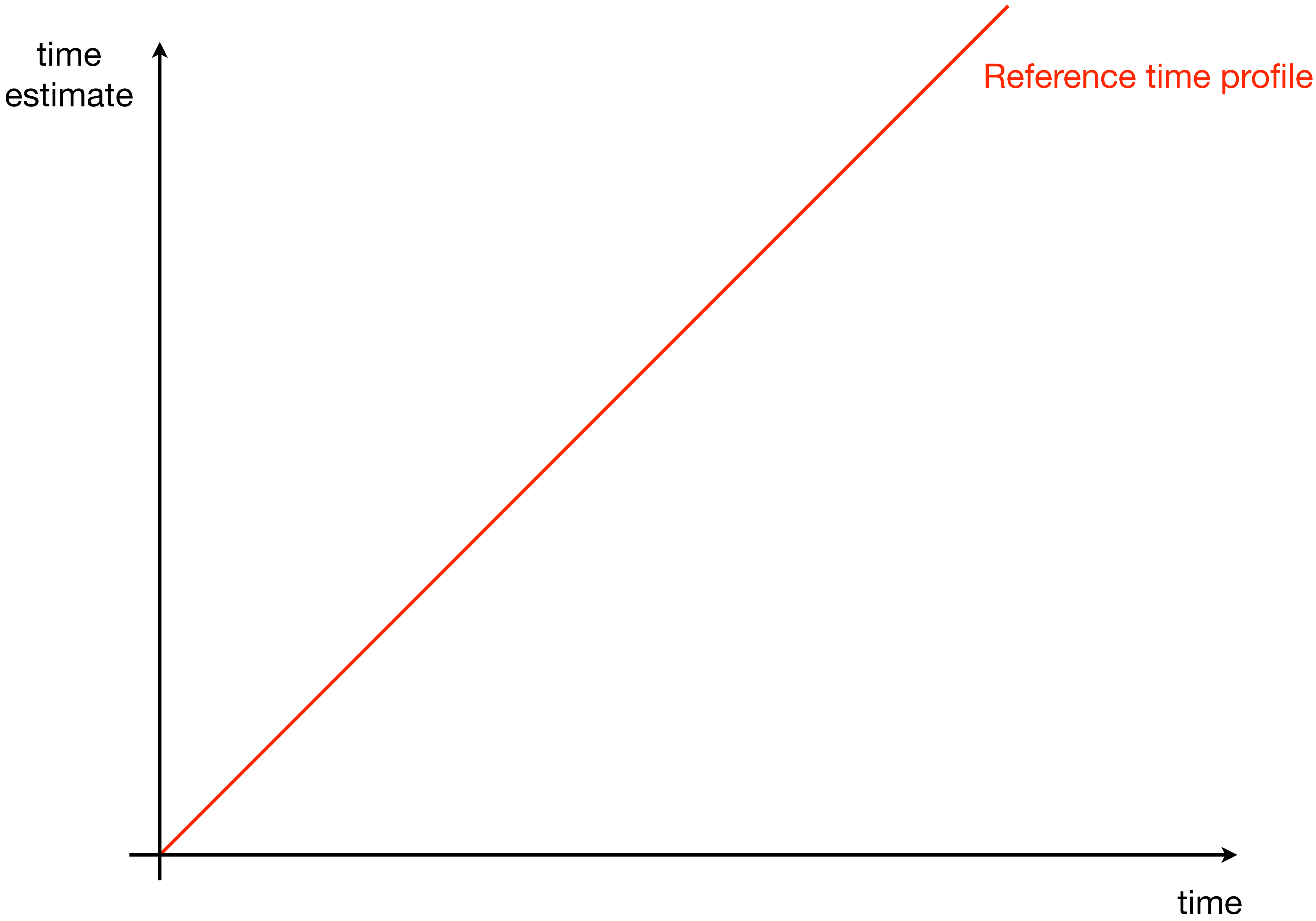
Clock synchronization



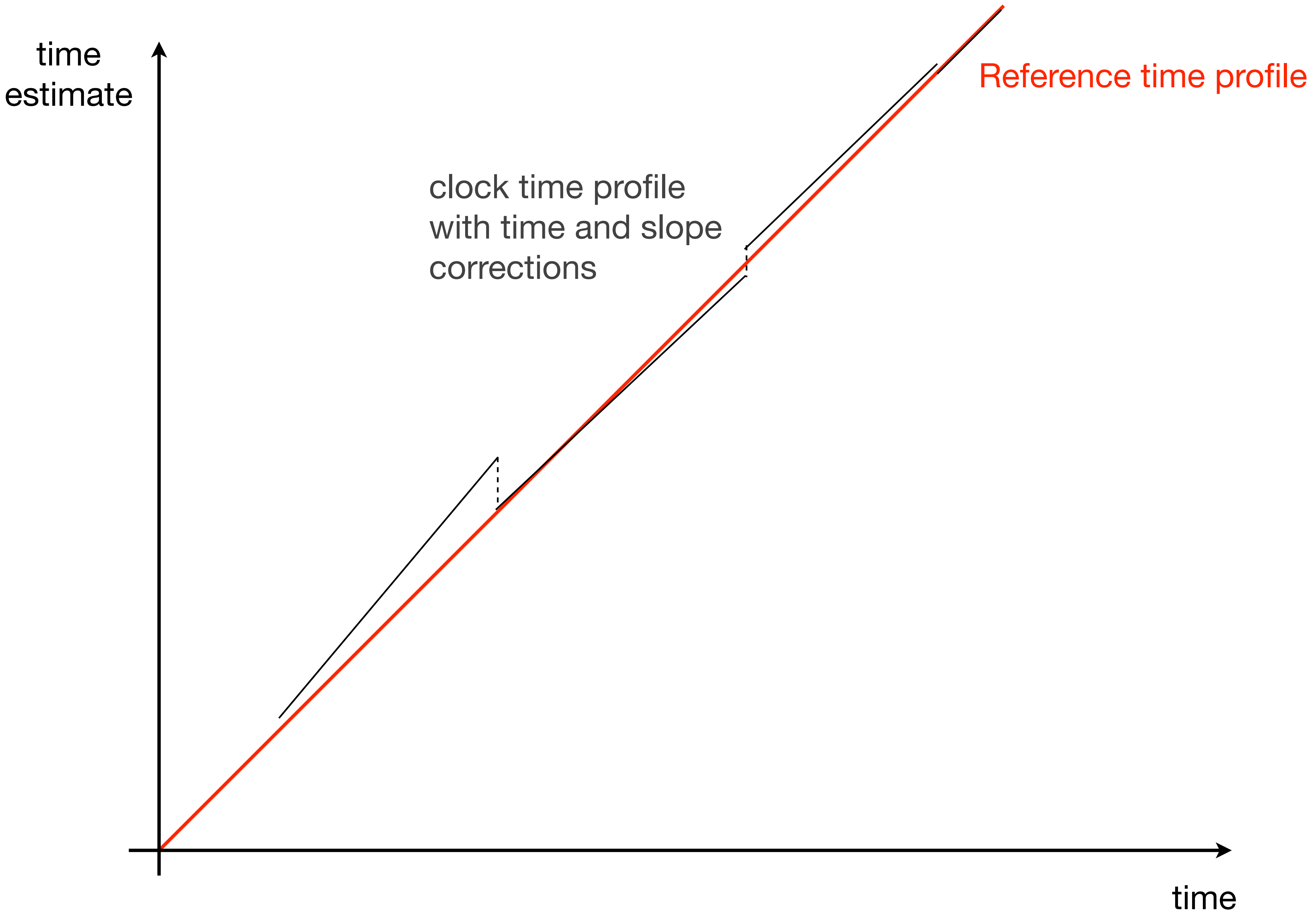
Clock synchronization



Clock synchronization

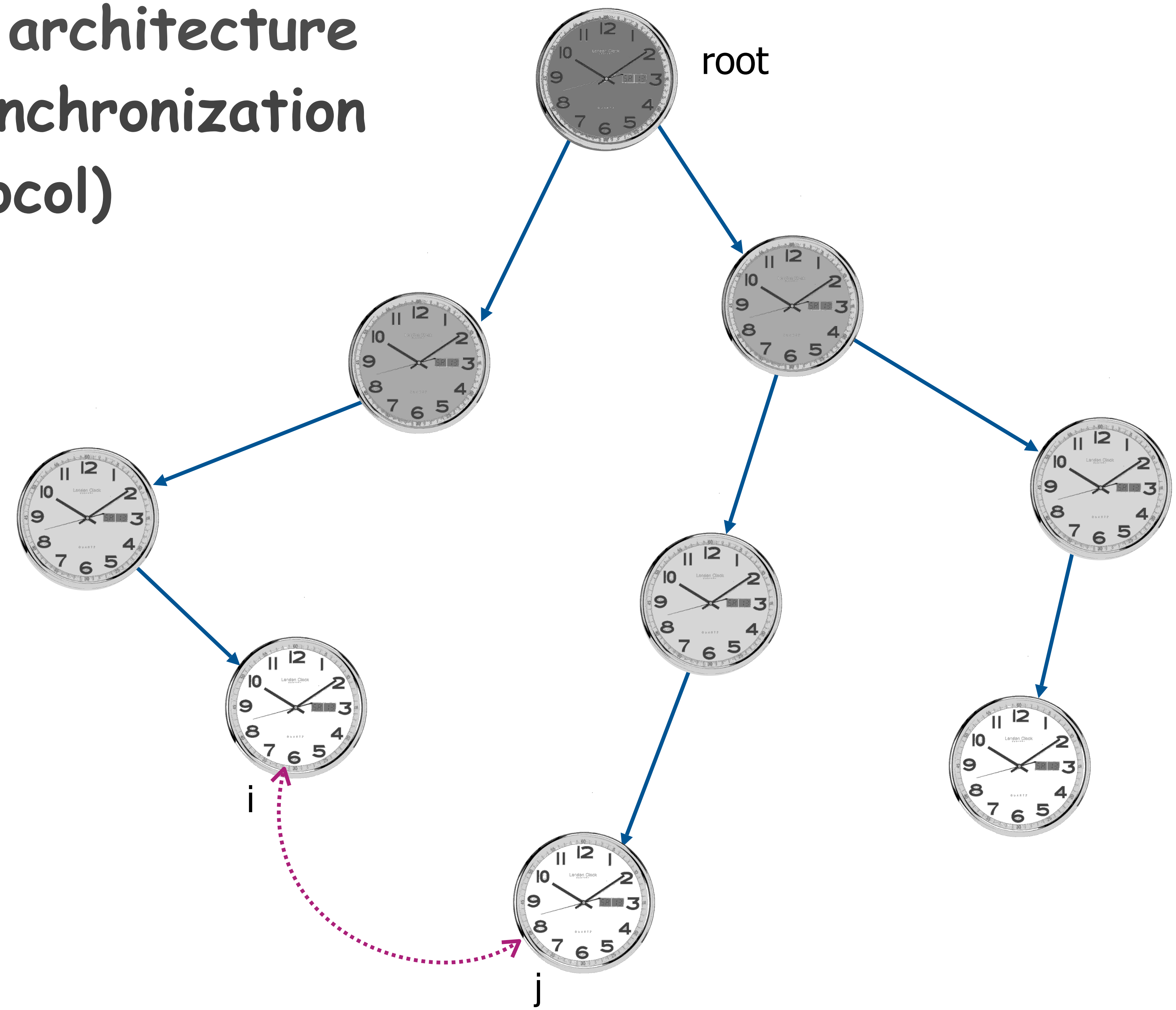


Clock synchronization

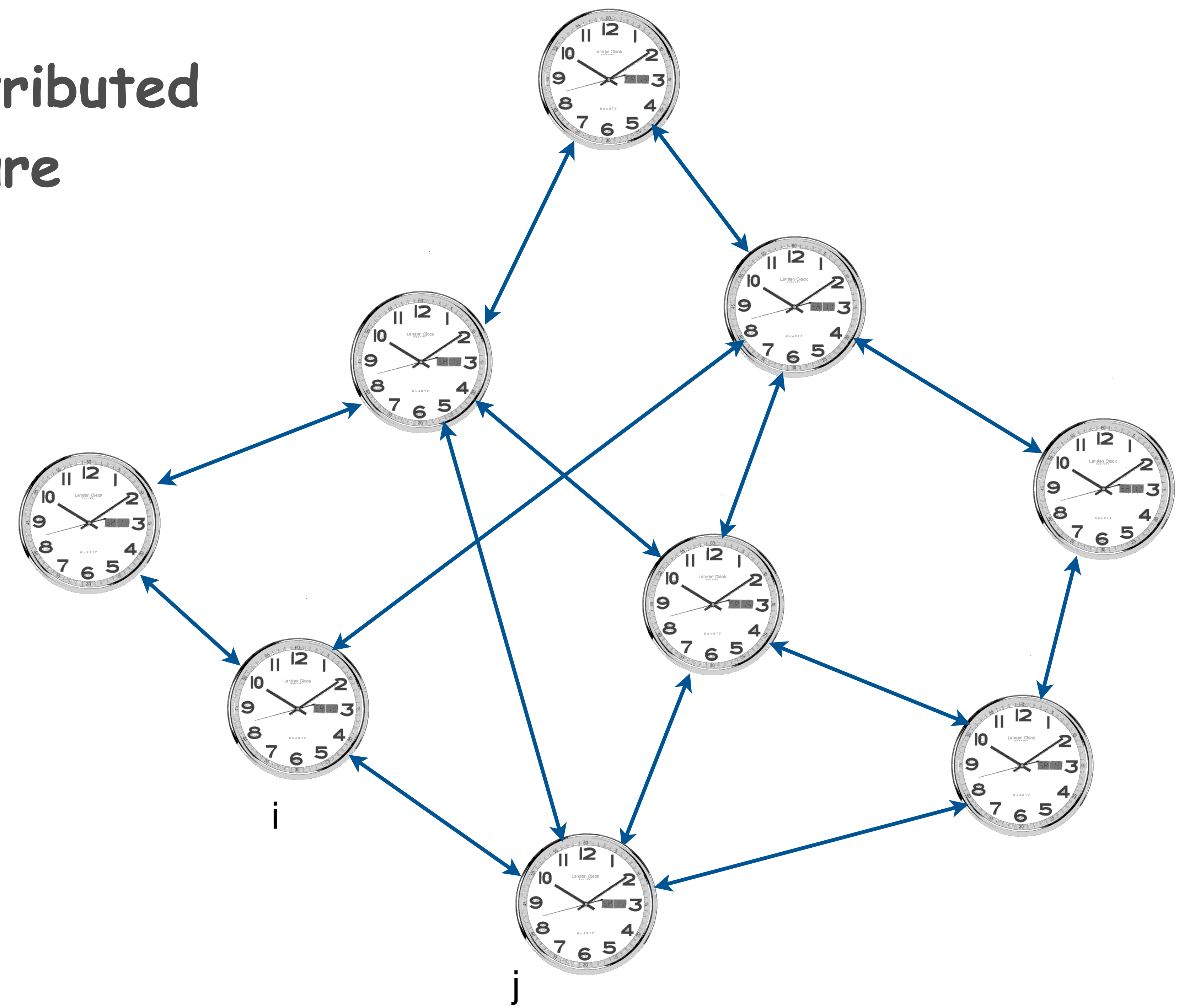


Architectures for clock synchronization

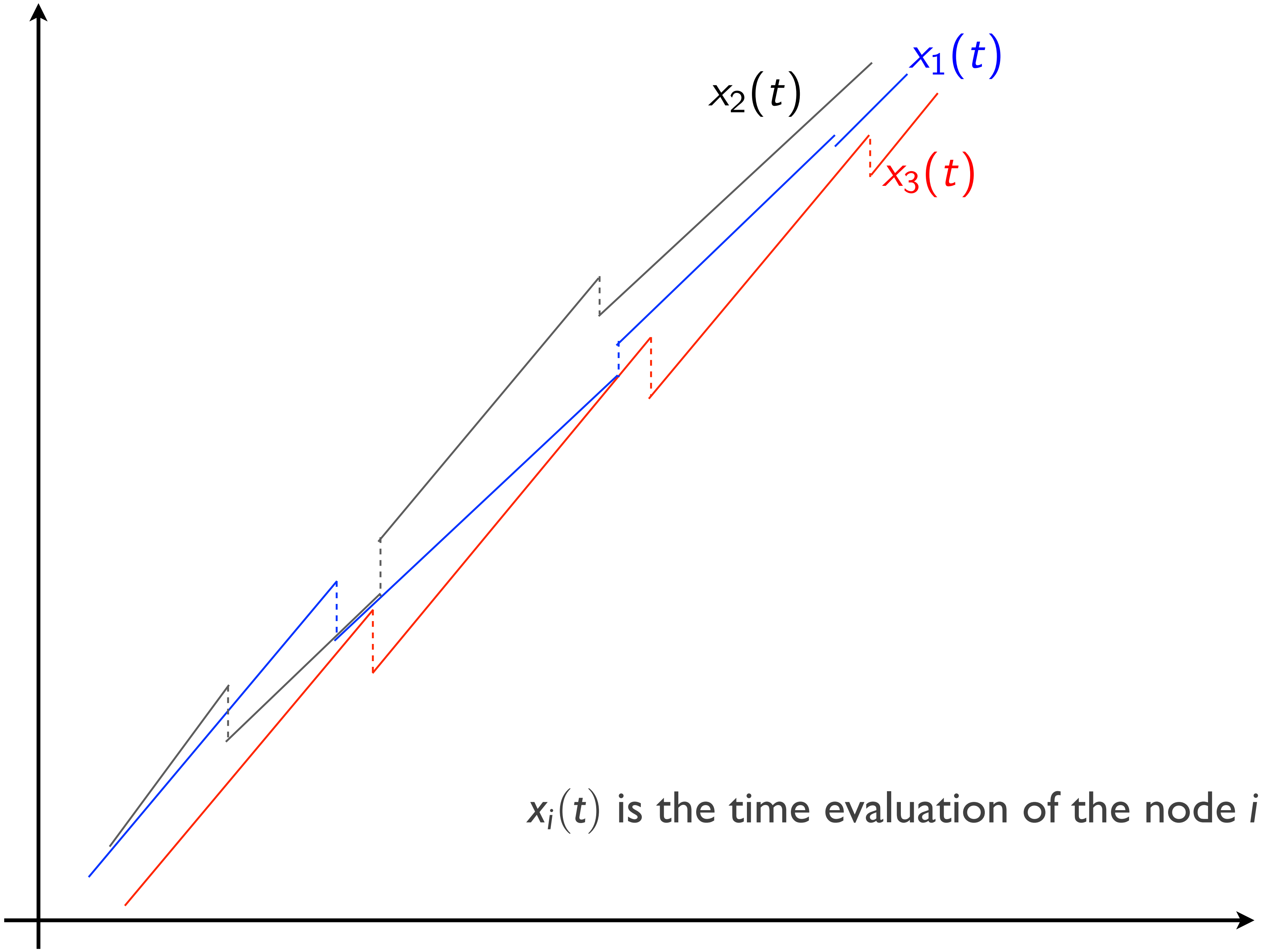
Master slave architecture (NTP time-synchronization protocol)



Leaderless distributed architecture



Clock synchronization



Clock synchronization with no reference time

Mathematical description of a clock

Each unit i has an oscillator with oscillator period Δ_i.

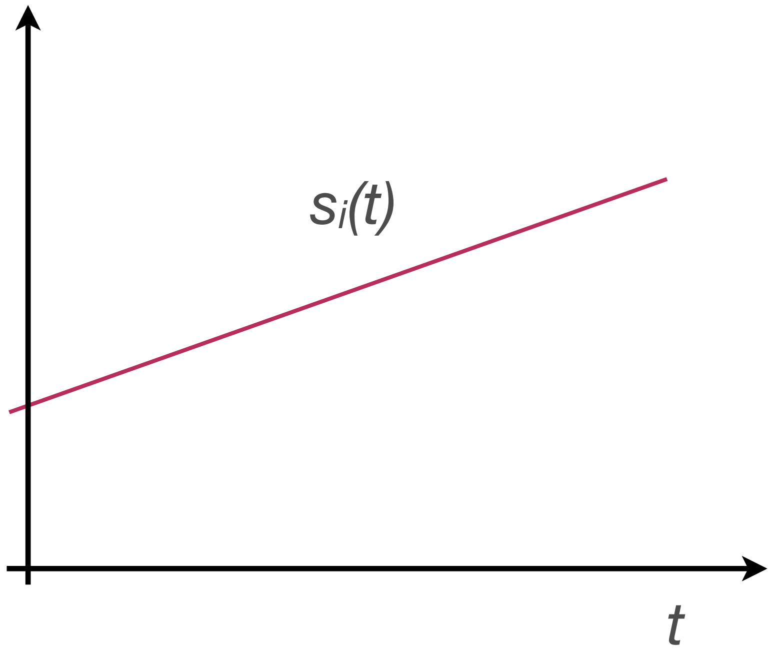
Mathematical description of a clock

Each unit i has an oscillator with oscillator period Δ_i.

From oscillator to counter

$$s_i(t) = \frac{t - t_0}{\Delta_i} + s_i(t_0)$$

We neglect the clock quantization



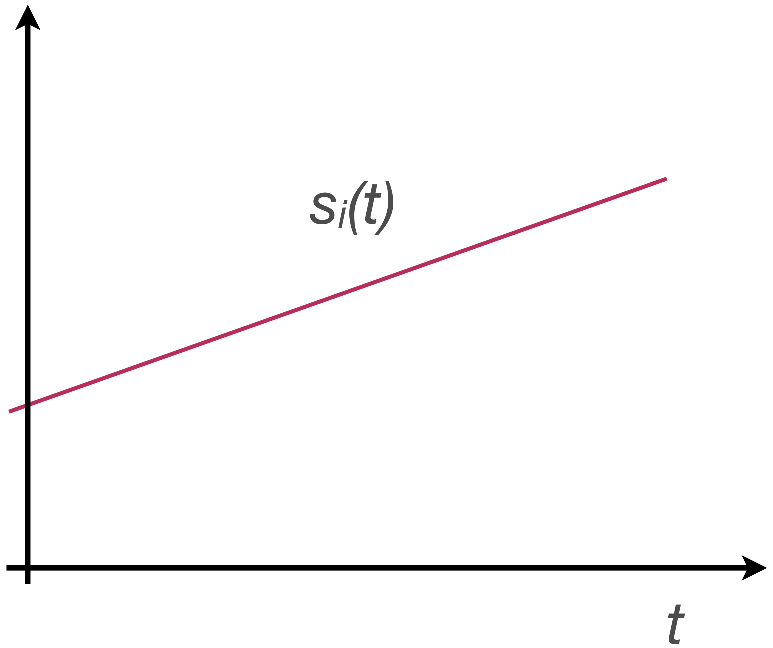
Mathematical description of a clock

Each unit i has an oscillator with oscillator period Δ_i.

From oscillator to counter

$$s_i(t) = \frac{t - t_0}{\Delta_i} + s_i(t_0)$$

We neglect the clock quantization



From counter to time estimator (clock)

$\hat{\Delta}_i$ is the an estimate of Δ_i

$$y_i(t) = y_i(t_0) + \hat{\Delta}_i(s_i(t) - s_i(t_0))$$

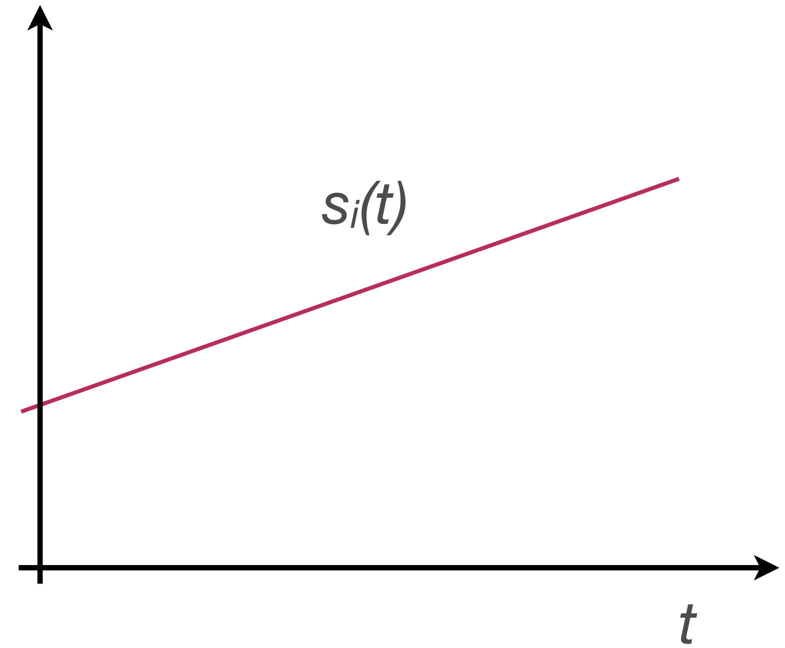
Mathematical description of a clock

Each unit i has an oscillator with oscillator period Δ_i.

From oscillator to counter

$$s_i(t) = \frac{t - t_0}{\Delta_i} + s_i(t_0)$$

We neglect the clock quantization



From counter to time estimator (clock)

$$y_i(t) \quad = \quad y_i(t_0) + \hat{\Delta}_i(s_i(t) - s_i(t_0))$$

$\hat{\Delta}_i$ is the an estimate of Δ_i

$\hat{\Delta}_i(t)$ We can have time dependent estimation of the oscillator period

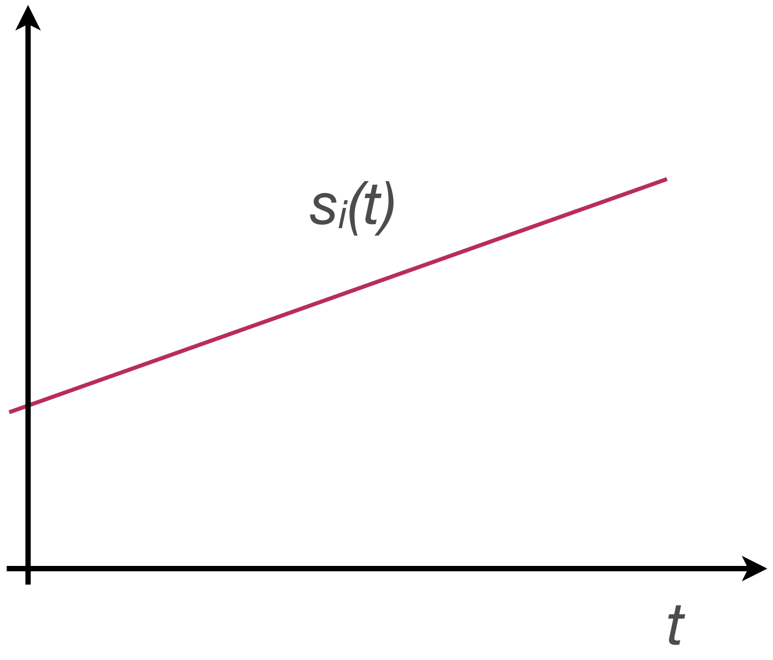
Mathematical description of a clock

Each unit i has an oscillator with oscillator period Δ_i.

From oscillator to counter

$$s_i(t) = \frac{t - t_0}{\Delta_i} + s_i(t_0)$$

We neglect the clock quantization



From counter to time estimator (clock)

$$\begin{aligned} y_i(t) &= y_i(t_0) + \hat{\Delta}_i(s_i(t) - s_i(t_0)) \\ &= y_i(t_0) + \hat{\Delta}_i \frac{t - t_0}{\Delta_i} \end{aligned}$$

$\hat{\Delta}_i$ is the an estimate of Δ_i

$\hat{\Delta}_i(t)$ We can have time dependent estimation of the oscillator period

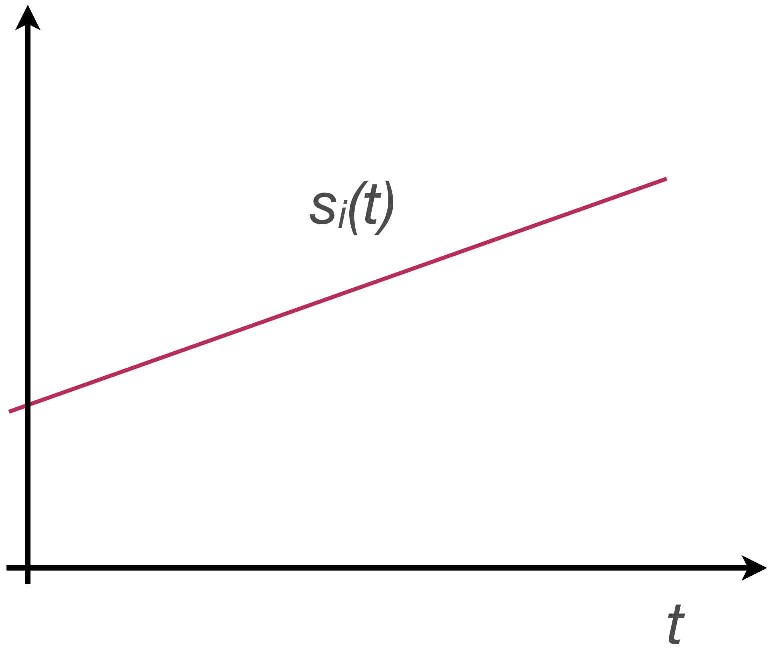
Mathematical description of a clock

Each unit i has an oscillator with oscillator period Δ_i.

From oscillator to counter

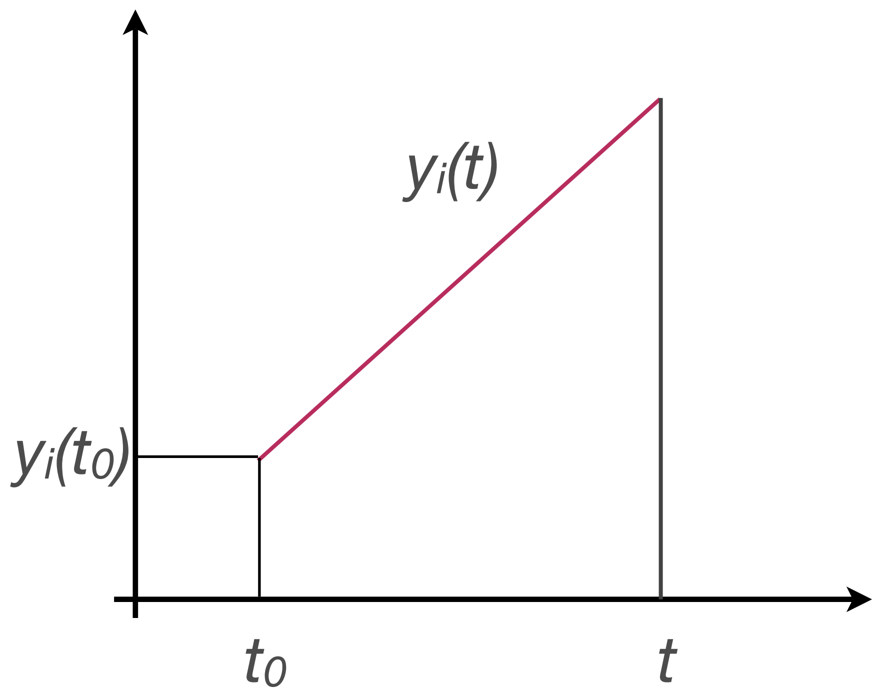
$$s_i(t) = \frac{t - t_0}{\Delta_i} + s_i(t_0)$$

We neglect the clock quantization



From counter to time estimator (clock)

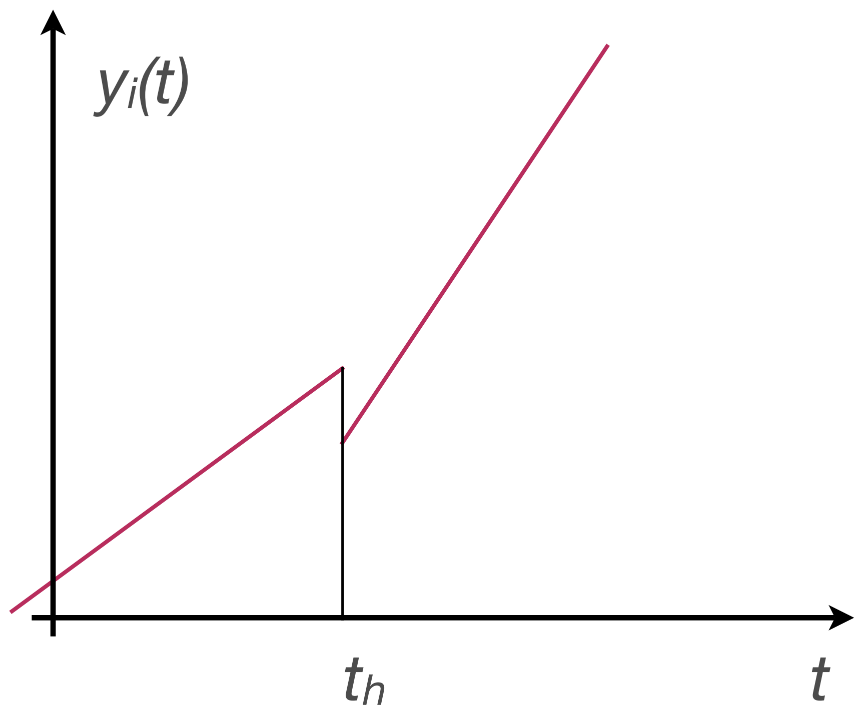
$$\begin{aligned} y_i(t) &= y_i(t_0) + \hat{\Delta}_i\left(s_i(t) - s_i(t_0)\right) \\ &= y_i(t_0) + \hat{\Delta}_i \frac{t - t_0}{\Delta_i} \end{aligned}$$



Mathematical description of a clock

Assume the units can operate corrections at time $t_1, t_2, t_3, \ldots$

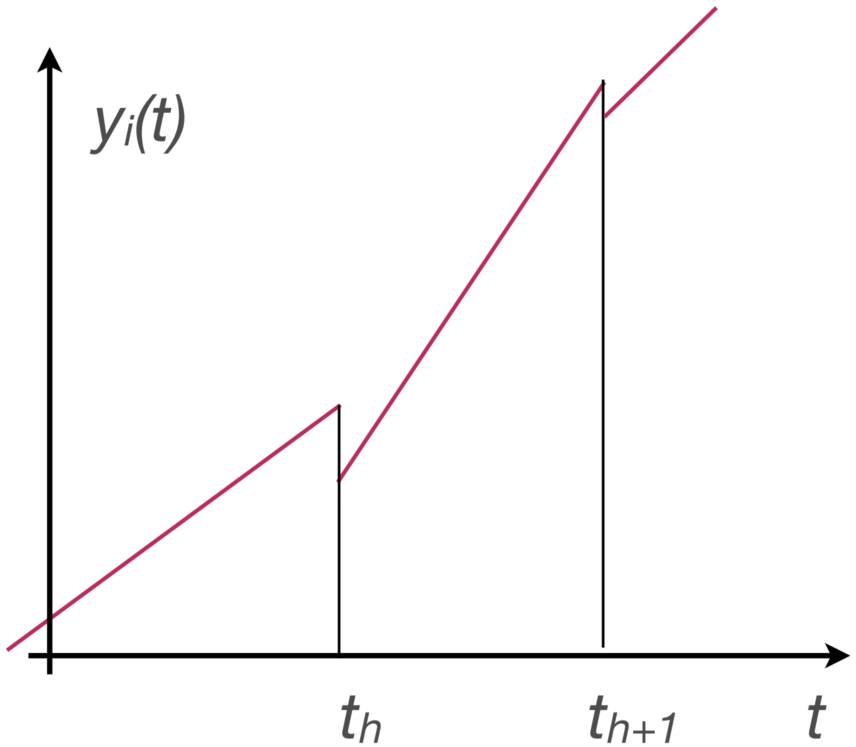
$$\begin{cases} y_i(t_h^+) = y_i(t_h^-) + u_i'(h) \\ \hat{\Delta}_i(t_h^+) = \hat{\Delta}_i(t_h^-) + u_i''(h) \end{cases}$$



Mathematical description of a clock

Remember that

$$y_i(t_{h+1}^-) = y_i(t_h^+) + \hat{\Delta}_i(t_h^+)\frac{t_{h+1} - t_h}{\Delta_i}$$

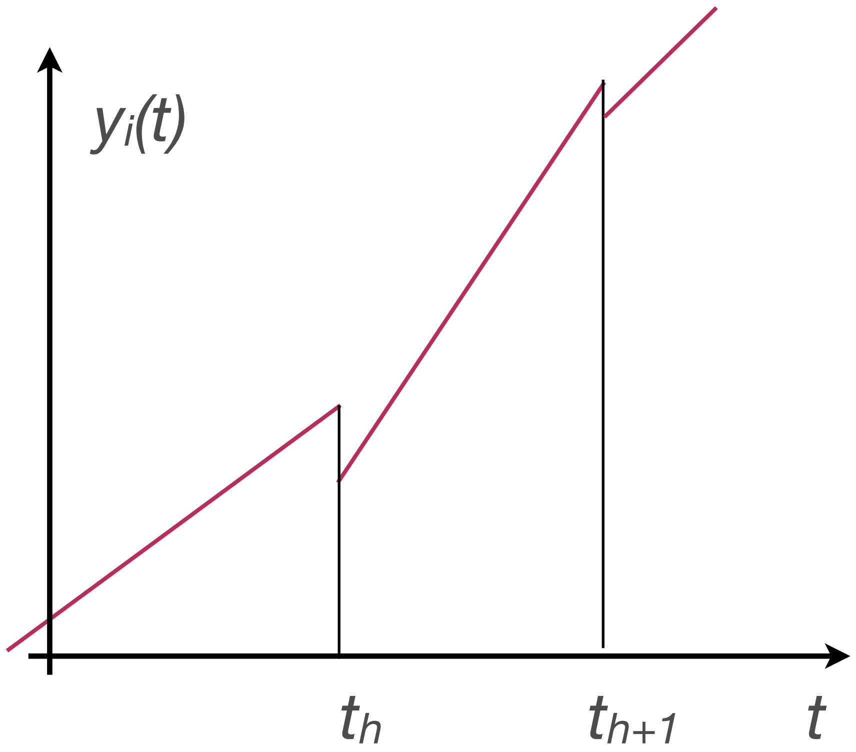


Mathematical description of a clock

Remember that

$$y_i(t_{h+1}^-) = y_i(t_h^+) + \hat{\Delta}_i(t_h^+)\frac{t_{h+1} - t_h}{\Delta_i}$$

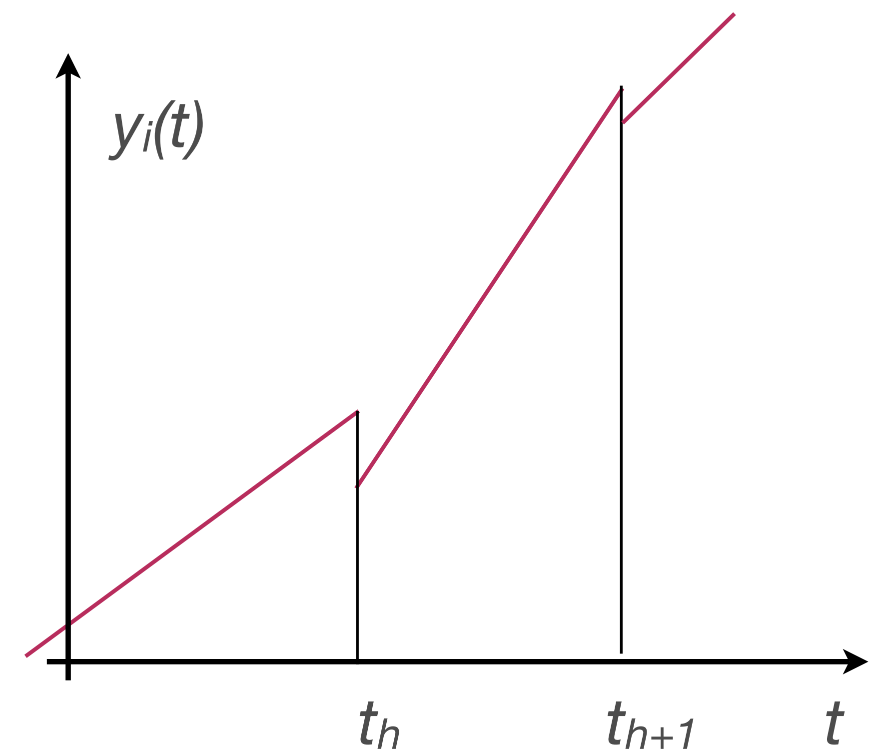
$$y_i(t_{h+1}^-) \quad = \quad y_i(t_h^-) + \frac{t_{h+1} - t_h}{\Delta_i}\hat{\Delta}_i(t_h^+) + u_i'(h)$$



Mathematical description of a clock

Remember that

$$y_i(t_{h+1}^-) = y_i(t_h^+) + \hat{\Delta}_i(t_h^+)\frac{t_{h+1} - t_h}{\Delta_i}$$

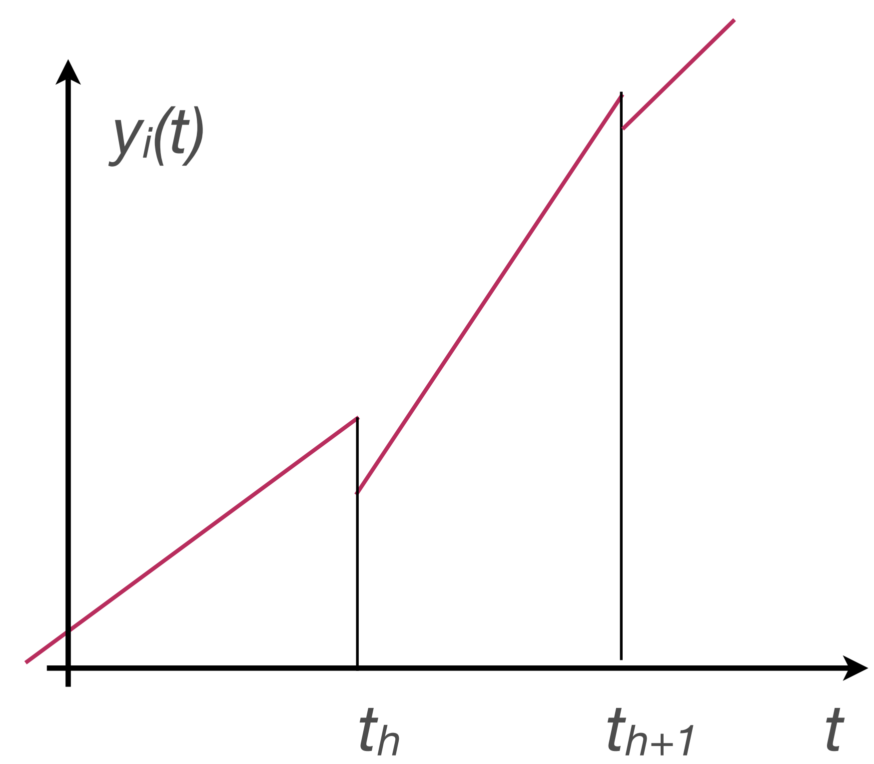


$$\begin{aligned} y_i(t_{h+1}^-) &= y_i(t_h^-) + \frac{t_{h+1} - t_h}{\Delta_i}\hat{\Delta}_i(t_h^+) + u_i'(h) \\ &= y_i(t_h^-) + \frac{t_{h+1} - t_h}{\Delta_i}(\hat{\Delta}_i(t_h^-) + u_i''(h)) + u_i'(h) \end{aligned}$$

Mathematical description of a clock

Remember that

$$y_i(t_{h+1}^-) = y_i(t_h^+) + \hat{\Delta}_i(t_h^+)\frac{t_{h+1} - t_h}{\Delta_i}$$



$$\begin{aligned}
y_i(t_{h+1}^-) &= y_i(t_h^-) + \frac{t_{h+1} - t_h}{\Delta_i}\hat{\Delta}_i(t_h^+) + u_i'(h) \\
&= y_i(t_h^-) + \frac{t_{h+1} - t_h}{\Delta_i}(\hat{\Delta}_i(t_h^-) + u_i''(h)) + u_i'(h)
\end{aligned}$$

Moreover

$$\hat{\Delta}_i(t_{h+1}^-) = \hat{\Delta}_i(t_h^-) + u_i''(h)$$

Mathematical description of a clock

Denoting

$$x_i(h) := \begin{bmatrix} y_i(t_h^-) \\ \hat{\Delta}_i(t_h^-) \end{bmatrix} \qquad u_i(h) := \begin{bmatrix} u_i'(h) \\ u_i''(h) \end{bmatrix}$$

we obtain

$$\begin{cases} x_i(h+1) = \begin{bmatrix} 1 & \frac{t_{h+1}-t_h}{\Delta_i} \\ 0 & 1 \end{bmatrix} (x_i(h) + u_i(h)) \\ \\ y_i(h) = \begin{bmatrix} 1 & 0 \end{bmatrix} x_i(h) \end{cases}$$

Consensus based clock synch

We propose the following linear control

$$u_i(h) = \sum_{j=1}^{N} K_{ij}(h)\, F\,(x_j(h) - x_i(h))$$

where

$$F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$$

is the local control matrix and $K_{ij}(h) \in \mathbb{R}$ are the interconnection coefficients forming the consensus matrix $K(h) \in \mathbb{R}^{N \times N}$.

Consensus based clock synch

We propose the following linear control

$$u_i(h) = \sum_{j=1}^{N} K_{ij}(h)\, F\, (x_j(h) - x_i(h))$$

where

$$F = \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}$$

is the local control matrix and $K_{ij}(h) \in \mathbb{R}$ are the interconnection coefficients forming the consensus matrix $K(h) \in \mathbb{R}^{N \times N}$.

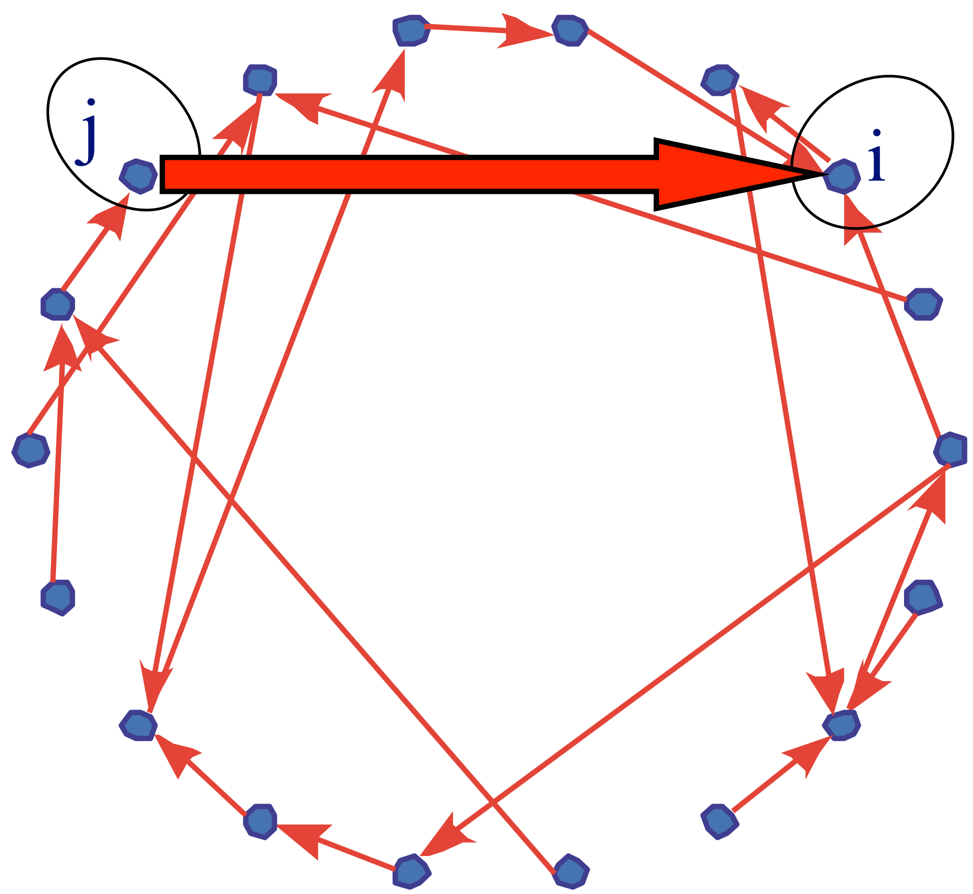
We need to design F and $K(h)$ such that

$$\boxed{|y_i(h) - y_j(h)| \to 0 \qquad \forall i, j}$$

Communication graph

At time t_h the node i needs to receive from the node j the state $x_j(h)$ only if $K_{ij}(h) \neq 0$.

$\mathcal{G}_{K(h)}$



Consensus for higher order systems

We get a consensus problem for higher order systems with two peculiarities:

- The dynamics are linear but time-varying.
- The dynamics of the systems are (slightly) different and partially unkown.

In the literature there are few theoretical results on this problem.

In case of time-varying topology

◇ *L. Scardovi, R. Sepulchre, "Synchronization in Networks of Identical Linear Systems", Automatica 45 (2009), 2557-2562*

In case of different systems

◇ *C.-Y. Kao, U.T. Jönsson, and H. Fujioka. "Characterization of robust stability of a class of interconnected systems". To appear in Automatica*

Ideal (unrealistic) time-invariant case

In the ideal unrealistic case:

- Regularly timed corrections $\boxed{t_h = hT + t_0;}$
- Time-invariant corrections $\boxed{K(h) = K.}$

We get

$$x_i(h+1) = \begin{bmatrix} 1 & T/\Delta_i \\ 0 & 1 \end{bmatrix} (x_i(h) + u_i(h))$$

$$u_i(h) = \sum_{j=1}^{N} K_{ij}\, F(x_j(h) - x_i(h))$$

We obtain a treatable consensus problem for non-identical systems with time invariant communication topology.

Ideal (unrealistic) time-invariant case

Consider the global system having the $2N$ dimensional state $x(h)$ formed by the states $x_i(h)$ and having the output $e(h)$ with components

$$e_i(h) := y_i(h) - \frac{1}{N}\sum_j y_j(h)$$

In the choice of K and F there are two constraints:

- ◇ The two modes associated with the eigenvalue 1 have to be non-observable.
- ◇ The other eigenvalues have to be inside the open unit circle.

Ideal (unrealistic) time-invariant case

Consider the global system having the $2N$ dimensional state $x(h)$ formed by the states $x_i(h)$ and having the output $e(h)$ with components

$$e_i(h) := y_i(h) - \frac{1}{N}\sum_j y_j(h)$$ Synchronization error

In the choice of K and F there are two constraints:

- ◇ The two modes associated with the eigenvalue 1 have to be non-observable.
- ◇ The other eigenvalues have to be inside the open unit circle.

Ideal (unrealistic) time-invariant case

The observability condition imposes that $f_{21} \neq 0$. Therefore the triangular form

$$F = \begin{bmatrix} f_{11} & f_{12} \\ 0 & f_{22} \end{bmatrix}$$

does not work (this is the solution proposed by Scardovi, Sepulchre for synchronizing double integrators).

Ideal (unrealistic) time-invariant case

The observability condition imposes that $f_{21} \neq 0$. Therefore the triangular form

$$\cancel{F = \begin{bmatrix} f_{11} & f_{12} \\ 0 & f_{22} \end{bmatrix}}$$

does not work (this is the solution proposed by Scardovi, Sepulchre for synchronizing double integrators).

Ideal (unrealistic) time-invariant case

The observability condition imposes that $f_{21} \neq 0$. Therefore the triangular form

$$\cancel{F = \begin{bmatrix} f_{11} & f_{12} \\ 0 & f_{22} \end{bmatrix}}$$

does not work (this is the solution proposed by Scardovi, Sepulchre for synchronizing double integrators).

The proposed solution is

$$F = \begin{bmatrix} f_{11} & 0 \\ f_{21} & 0 \end{bmatrix}$$

Ideal (unrealistic) time-invariant case

The observability condition imposes that $f_{21} \neq 0$. Therefore the triangular form

$$\cancel{F = \begin{bmatrix} f_{11} & f_{12} \\ 0 & f_{22} \end{bmatrix}}$$

does not work (this is the solution proposed by Scardovi, Sepulchre for synchronizing double integrators).

The proposed solution is

$$F = \begin{bmatrix} f_{11} & 0 \\ f_{21} & 0 \end{bmatrix}$$

$$\boxed{u_i(h) = \sum_{j=1}^{N} K_{ij} \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} (y_j(h) - y_i(h))}$$

Ideal (unrealistic) time-invariant case

Solution

◇ If the graph G describing the feasible communications is undirected and Δ_i's are equal, then

$$\boxed{\begin{gathered} f_{11} = 1/3, \qquad f_{21} \leq \Delta/T \\ K_{ij} = \begin{cases} -\frac{1}{\max\{d_i, d_j\}} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j \\ -\sum_{j \neq i} K_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \end{gathered}}$$

where d_i denotes the number of neighbors of the node i.

◇ If Δ_i's are not equal, then for continuity the same choice works also if Δ_i are slightly different. We can use H^∞ methods to evaluate how different these Δ_i's can be.

◇ If G is not undirected, then ???

Ideal (unrealistic) time-invariant case

Solution

◇ If the graph G describing the feasible communications is undirected and Δ_i's are equal, then

$$f_{11} = 1/3, \qquad f_{21} \leq \Delta/T$$

$$K_{ij} = \begin{cases} -\frac{1}{\max\{d_i, d_j\}} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j \\ -\sum_{j \neq i} K_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Metropolis weights

where d_i denotes the number of neighbors of the node i.

◇ If Δ_i's are not equal, then for continuity the same choice works also if Δ_i are slightly different. We can use H^∞ methods to evaluate how different these Δ_i's can be.

◇ If G is not undirected, then ???

Clock synchronization

Simulations shows the same algorithm yields synchronization also in the general time-varying case and works well even in presence of time-varying oscillator period and communication delays. However we have not been able to prove this mathematically.

We could prove the local asymptotic synchronization of the so-called pseudo-synchronous implementation.

Conclusions

- The consensus algorithm is an instance of a completely distributed design. This is an extreme design paradigm.
- It is intrinsically robust to external changes and highly self-adaptive so that a limited initial configuration and tuning effort is necessary.
- None or limited information about the global structure of the system is necessary to the units.
- Graceful performance degradation.
- Importance of the interaction network topology.

Conclusions

There are two important messages:

- The consensus algorithm should be analyzed in the context of the applications in which it is used. This yields different performance indices with different relations with network topology.
- In large scale networks both time and the number of agents may be large. Therefore there might emerge several asymptotic regimes in relation to how these two quantities grow with respect to each other.