Multi-Agent Systems

Kai Cai

cai@omu.ac.jp

Last week

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i has an initial value $x_i(0)$

Distributed algorithm

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

$$(a_{ij} > 0, \sum_{j \in \mathcal{N}_i} a_{ij} < 1)$$

$$x(k+1) = (I - L)x(k) \quad x := \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Last week

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i has an initial value $x_i(0)$

$x(k+1) = (I - L)x(k)$ solves averaging

$$\text{i.e. } (\forall v_i \in \mathcal{V}) \lim_{k \to \infty} x_i(k) = \frac{1}{n} \sum_{v_i \in \mathcal{V}} x_i(0)$$

iff $\mathcal{G}$ is strongly connected and weight balanced

Last week

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i has a local cost function

quadratic $f_i(x) = x^2 + \alpha_i x + \beta_i$

system's global cost function

$$F(x) := \frac{1}{n} \sum_{v_i \in \mathcal{V}} f_i(x)$$

which has a unique optimal solution x^*

Problem: update $x_i(k)$, $k = 1, 2, \ldots,$

s.t. $(\forall v_i \in \mathcal{V}) \lim_{k \to \infty} x_i(k) = x^*$

Last week

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i has a local cost function

quadratic $f_i(x) = x^2 + \alpha_i x + \beta_i$

Distributed algorithm

$$x(k+1) = (I - L)x(k) \qquad x := \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

with $x(0) = [-\frac{\alpha_1}{2} \ \cdots \ -\frac{\alpha_n}{2}]^\top$

Last week

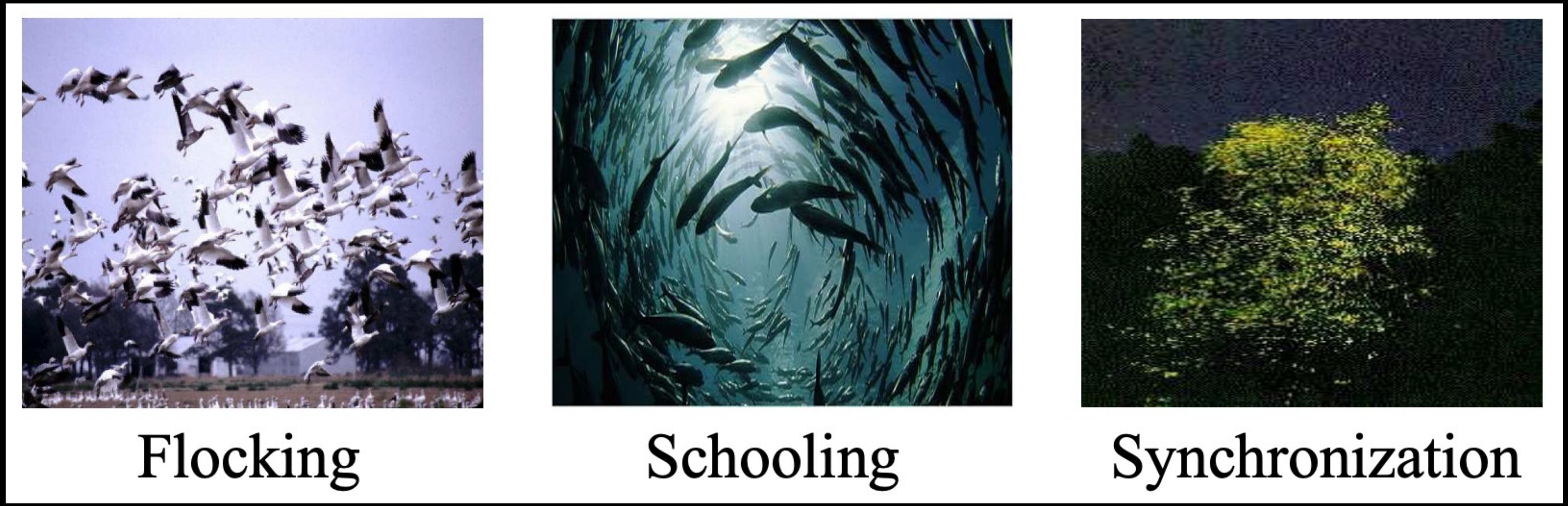
a system of n interacting agents
is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
each agent v_i has a local cost function
quadratic $f_i(x) = x^2 + \alpha_i x + \beta_i$

$x(k+1) = (I - L)x(k)$ with initial
condition $x(0) = [-\frac{\alpha_1}{2} \ \cdots \ -\frac{\alpha_n}{2}]^\top$ solves
optimization , i.e. $(\forall v_i \in \mathcal{V}) \lim_{k \to \infty} x_i(k) = x^*$

iff $\mathcal{G}$ is strongly connected and
weight balanced

Consensus/ Rendezvous

Examples



Multi-agent system

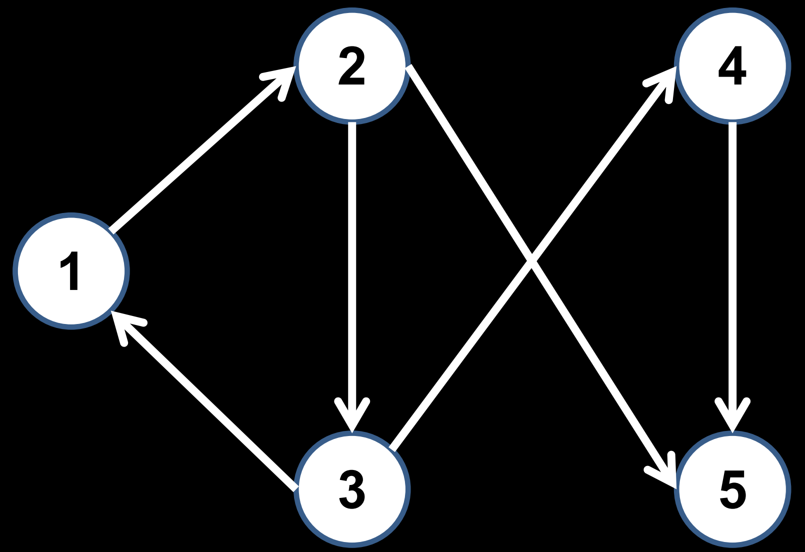
a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

node $v_i \in \mathcal{V}$: an agent

edge $(v_j, v_i) \in \mathcal{E}$: agent j sends information to v_i

example:



Consensus problem

each agent v_i updates its state based on

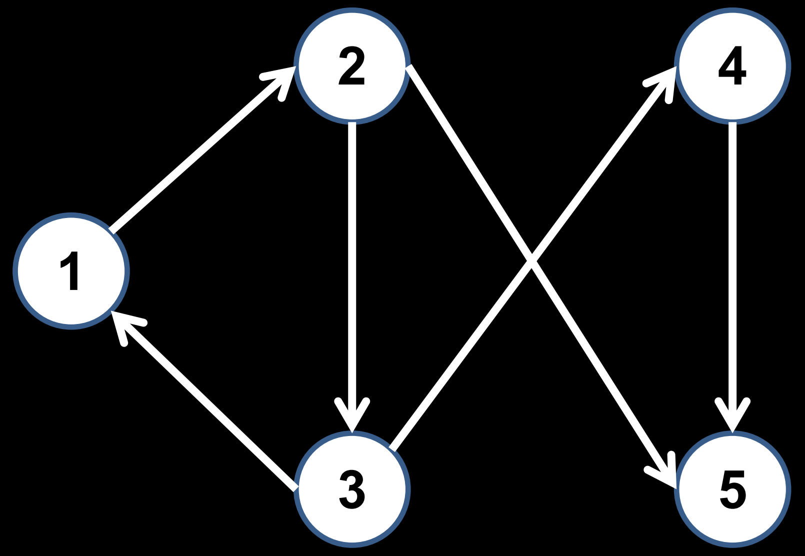
$$\dot{x}_i = u_i, \; x_i, u_i \in \mathbb{R}$$

consensus: design input $u_i(t)$, $t \geq 0$

s.t. $(\forall x_1(0), \ldots, x_n(0))(\exists c \in \mathbb{R}) x_i(t) \to c$

as $t \to \infty$

example:



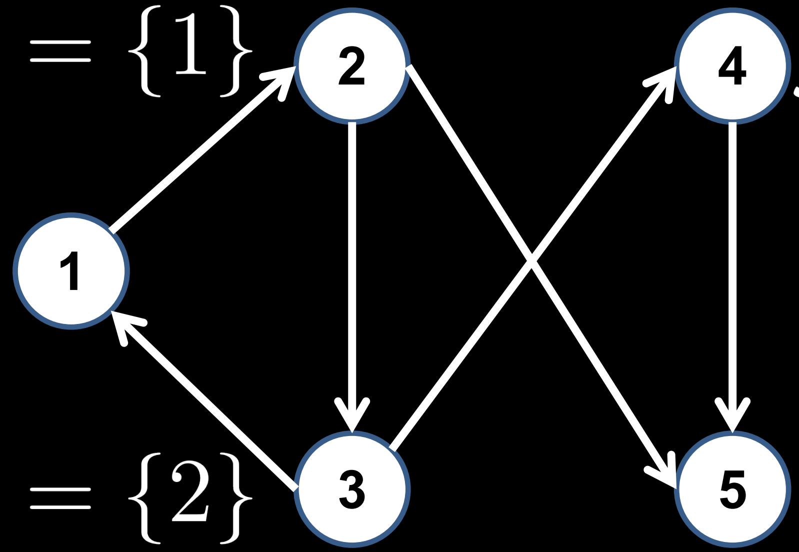
Distributed algorithm

each agent v_i can receive information $x_j(t)$ from neighbor(s) $j \in \mathcal{N}_i$

distributed algorithm: at time $t(\geq 0)$
design $u_i(t)$ based on information $x_j(t)$
where $j \in \mathcal{N}_i$

example: $\mathcal{N}_2 = \{1\}$ $\mathcal{N}_4 = \{3\}$

$\mathcal{N}_1 = \{3\}$



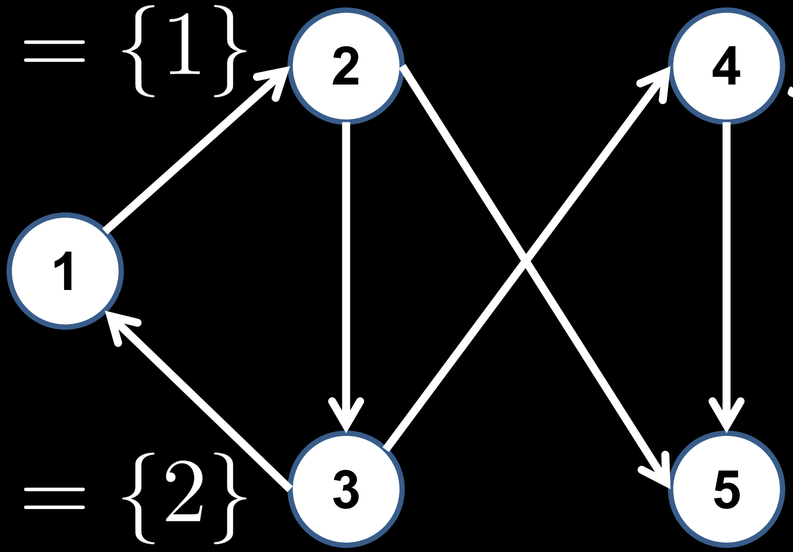
$\mathcal{N}_3 = \{2\}$ $\mathcal{N}_5 = \{2, 4\}$

Example

$$
\begin{aligned}
\dot{x}_1 &= u_1 = x_3 - x_1 \\
\dot{x}_2 &= u_2 = x_1 - x_2 \\
\dot{x}_3 &= u_3 = x_2 - x_3 \\
\dot{x}_4 &= u_4 = x_3 - x_4 \\
\dot{x}_5 &= u_5 = (x_2 - x_5) + (x_4 - x_5)
\end{aligned}
$$

example: $\mathcal{N}_2 = \{1\}$ $\mathcal{N}_4 = \{3\}$

$\mathcal{N}_1 = \{3\}$



$\mathcal{N}_5 = \{2, 4\}$

$\mathcal{N}_3 = \{2\}$

Example

$$\dot{x}_1 = u_1 = x_3 - x_1$$

$$\dot{x}_2 = u_2 = x_1 - x_2$$

$$\dot{x}_3 = u_3 = x_2 - x_3$$

$$\dot{x}_4 = u_4 = x_3 - x_4$$

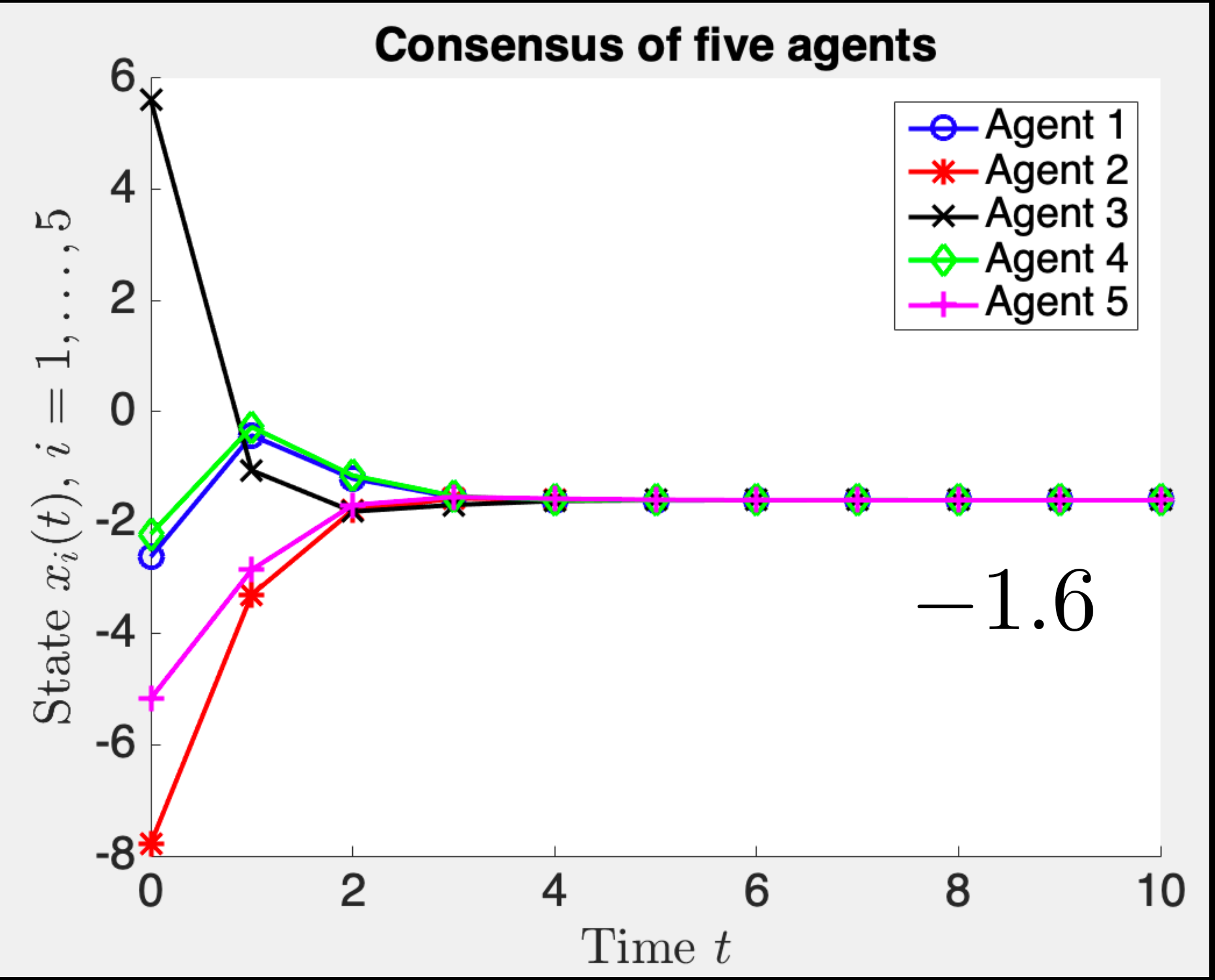
$$\dot{x}_5 = u_5 = (x_2 - x_5) + (x_4 - x_5)$$

$$\dot{x}_i = u_i = \sum_{j \in \mathcal{N}_i} (x_j - x_i)$$

relative state information

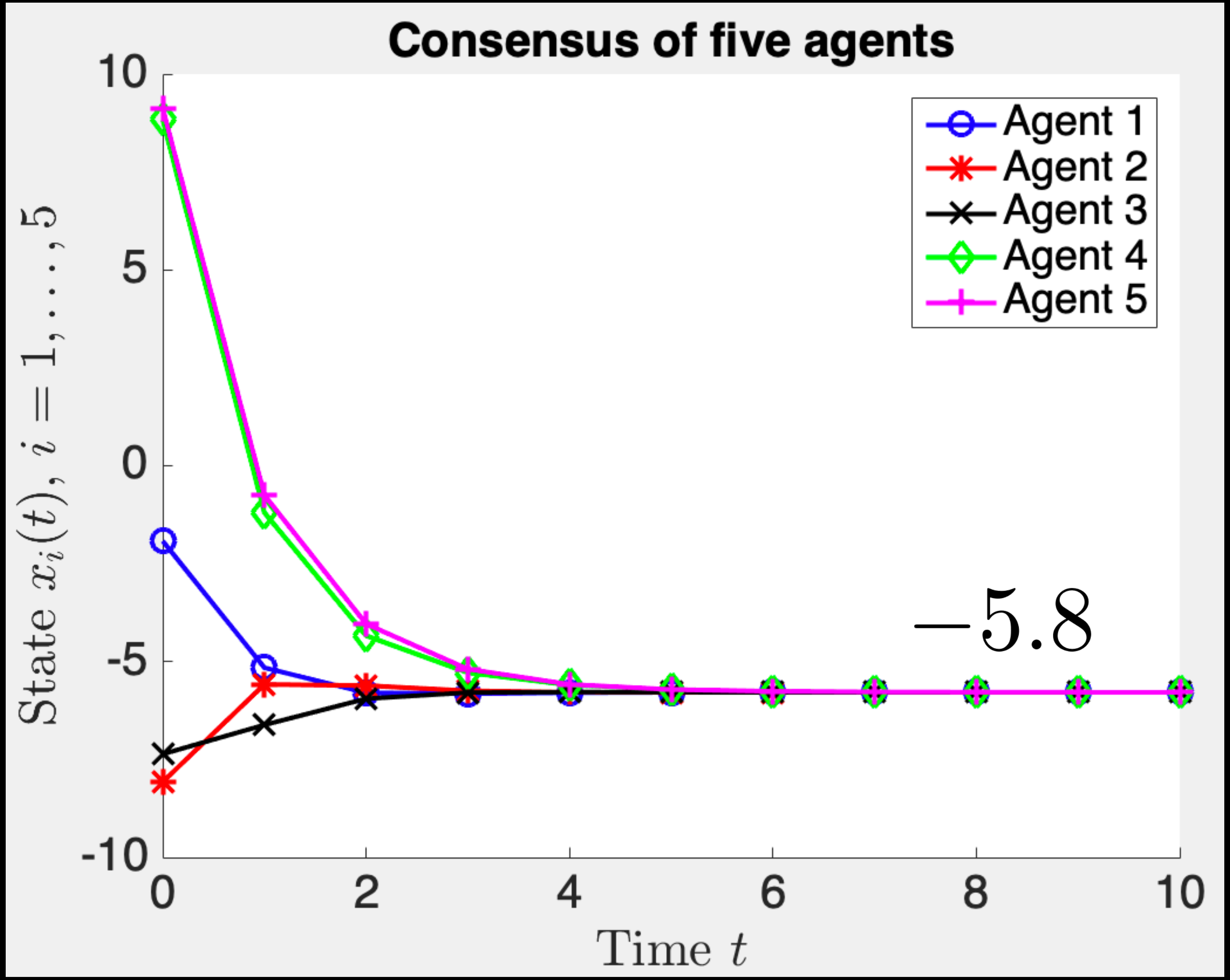
Example

simulation: $x_1(0) = -2.6, x_2(0) = -7.8$
$x_3(0) = 5.6, x_4(0) = 2.2, x_5(0) = -5.2$



Example

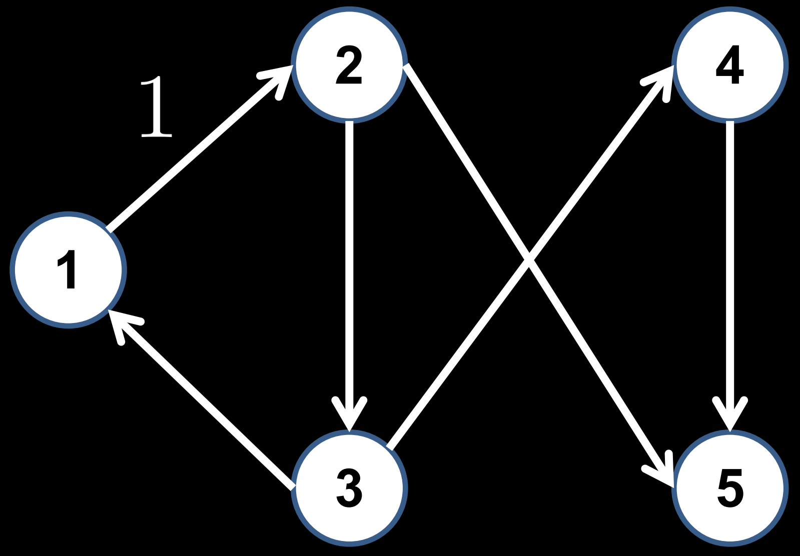
$$\text{simulation: } x_1(0) = -1.9, x_2(0) = -8.1$$
$$x_3(0) = -7.4, x_4(0) = 8.8, x_5(0) = 9.1$$



Weighted graph

example:

weighted graph $\mathcal{G}$



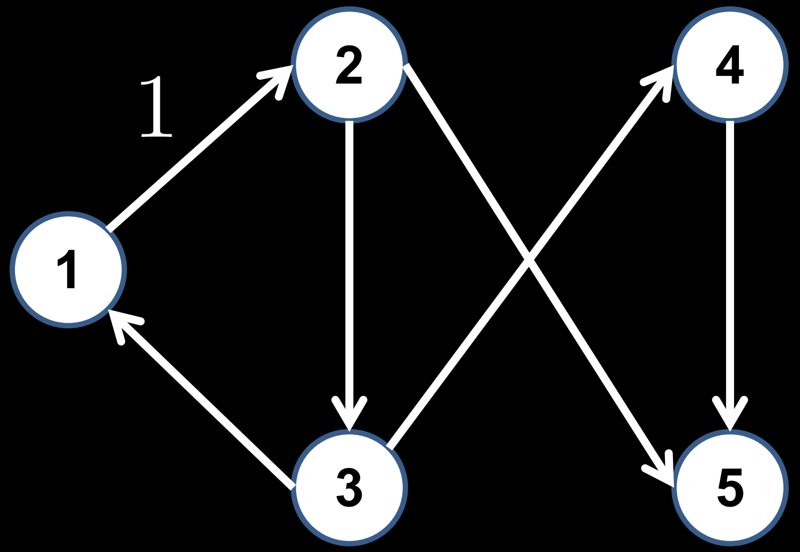
adjacency matrix

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

Weighted graph

example:

weighted graph $\mathcal{G}$



Laplacian matrix

$$L = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Equation

$$\dot{x}_1 = u_1 = x_3 - x_1$$
$$\dot{x}_2 = u_2 = x_1 - x_2$$
$$\dot{x}_3 = u_3 = x_2 - x_3$$
$$\dot{x}_4 = u_4 = x_3 - x_4$$
$$\dot{x}_5 = u_5 = (x_2 - x_5) + (x_4 - x_5)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 & -2 \end{bmatrix}}_{-L} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}$$

Recap, generalization

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i updates its state based on

$$\dot{x}_i = u_i, \;\; x_i, u_i \in \mathbb{R}$$

Problem: design u_i to update x_i

s.t. $(\forall v_i \in \mathcal{V})(\forall x_i(0))(\exists c) \lim_{t \to \infty} x_i(t) = c$

Recap, generalization

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i updates its state based on

$$\dot{x}_i = u_i,\ x_i, u_i \in \mathbb{R}$$

Distributed algorithm

$\dot{x}_i = u_i = \sum_{j \in \mathcal{N}_i} (x_j - x_i)$

based on information $x_j(t)$ or

relative information $x_j(t) - x_i(t)$

from neighbor agent(s) $j \in \mathcal{N}_i$

Recap, generalization

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i updates its state based on

$$\dot{x}_i = u_i, \;\; x_i, u_i \in \mathbb{R}$$

$$x := \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$\dot{x} = -Lx$$

Theorem

a system of n interacting agents

is modeled by graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

each agent v_i updates its state based on

$$\dot{x}_i = u_i, \; x_i, u_i \in \mathbb{R}$$

$\dot{x} = -Lx$ solves consensus

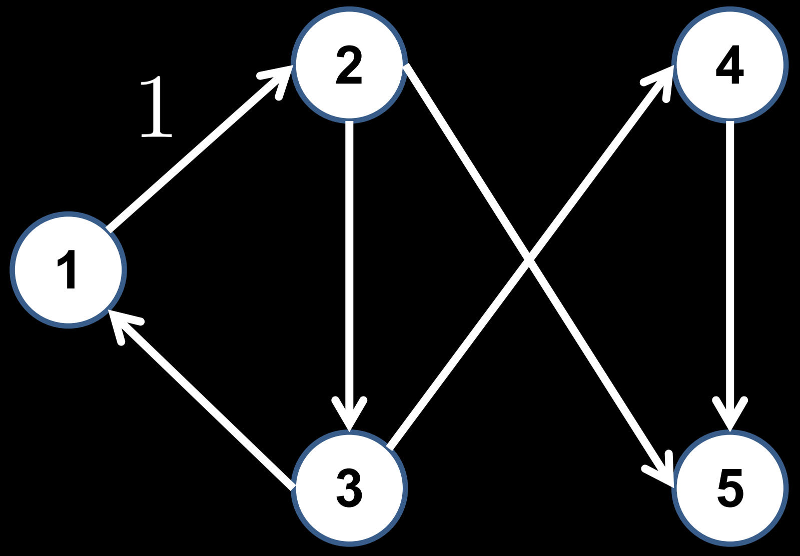
s.t. $(\forall v_i \in \mathcal{V})(\forall x_i(0))(\exists c) \lim_{t \to \infty} x_i(t) = c$

iff $\mathcal{G}$ contains a spanning tree

Example

example:

weighted graph $\mathcal{G}$



spanning tree (?)