Consensus Algorithms for Optimization in Multi-Agent Networks

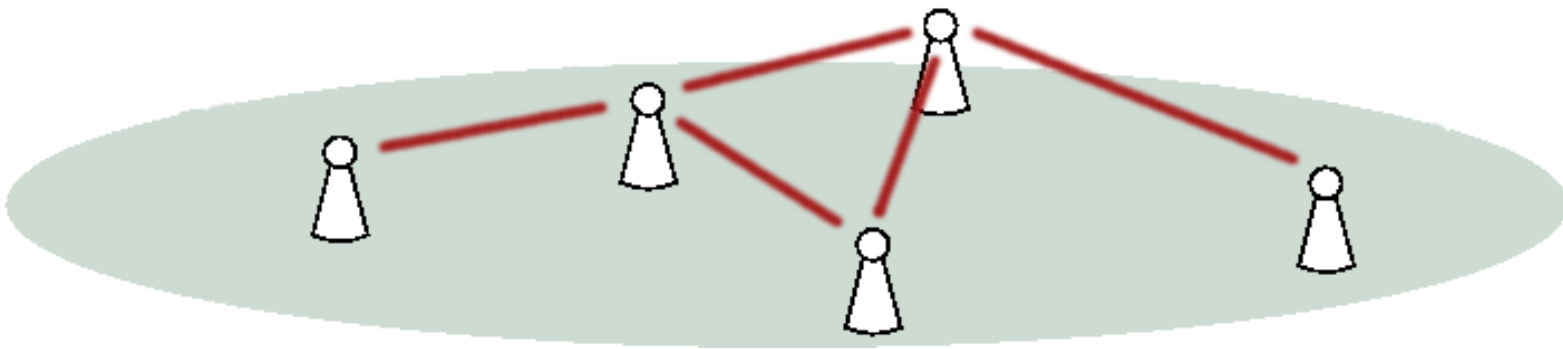
Peyresq'13

Pascal Bianchi

June 28, 2013

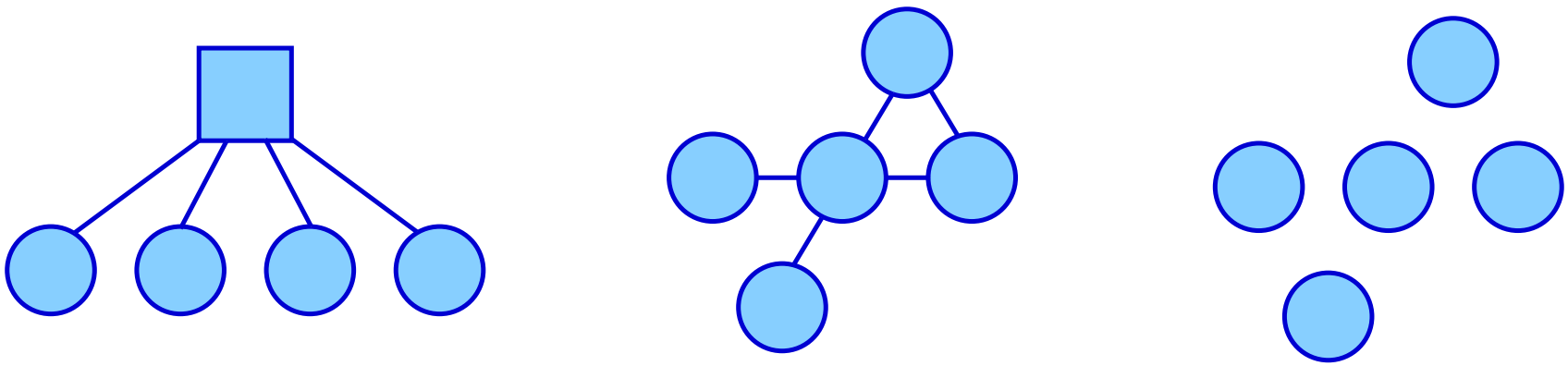
Context

Consider a network composed of N agents



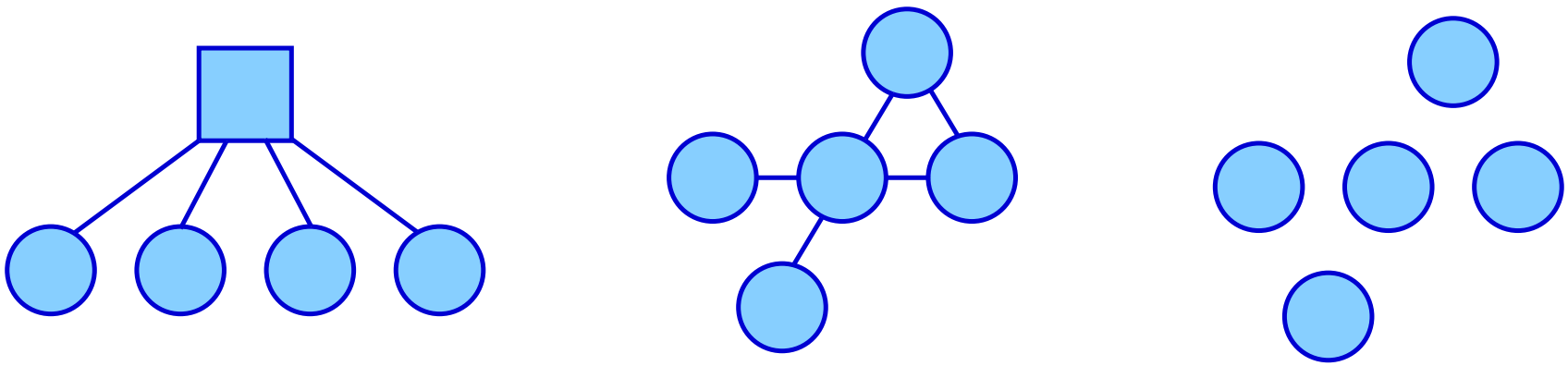
- Agents process *local* data
- Agents cooperate to estimate some *global* parameter

Network architectures



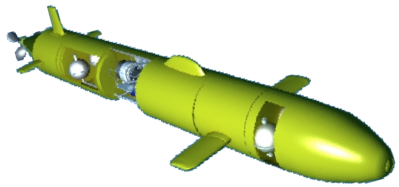
- **Centralized** : A node (reducer, sink) aggregates the agents' outputs
- **Distributed** : No central node - Agents cooperate with their neighbors
- **Non cooperative** : Agents are players who don't share information

Network architectures



- **Centralized** : A node (reducer, sink) aggregates the agents' outputs
- **Distributed** : No central node - Agents cooperate with their neighbors
- **Non cooperative** : Agents are players who don't share information

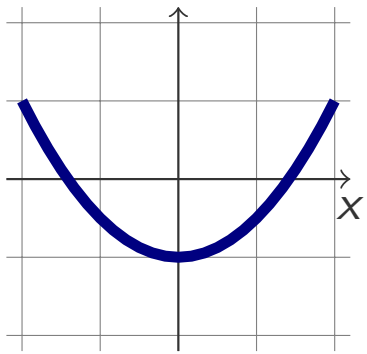
Some examples

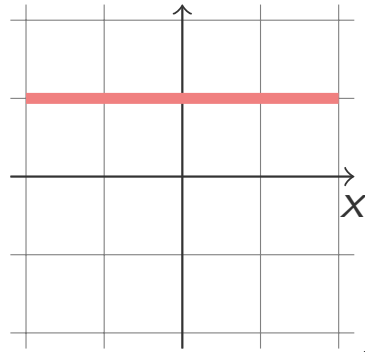
Network	*Agent*	*Objectives*
Ad-hoc network	Mobile terminal 	• Power control • Load balancing • Self-localization
Flotilla	Autonomous Underwater Vehicle 	• Trajectory planning • Flocking • Localization and mapping
Cloud	Virtual machine 	• Regression on distributed data sets • Distributed clustering

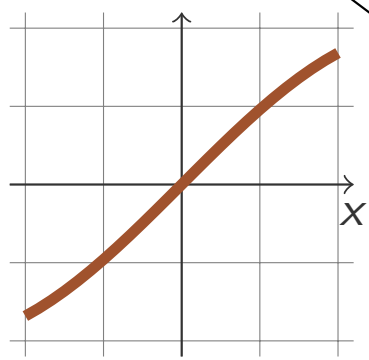
Outline

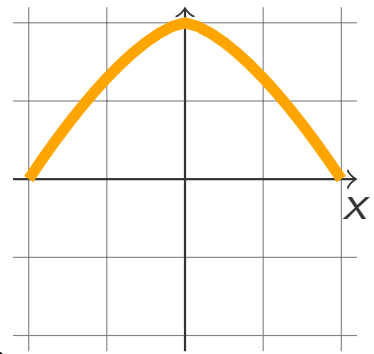
Outline

The Problem

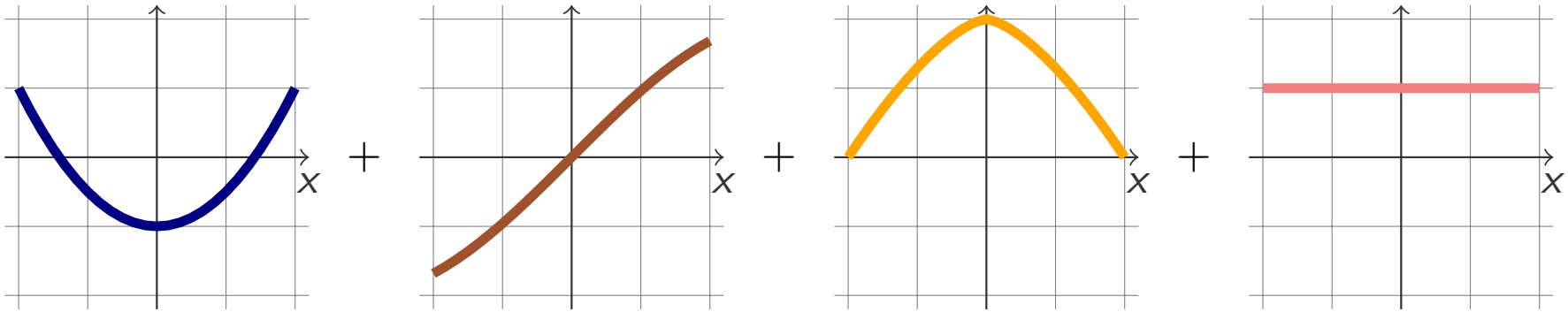




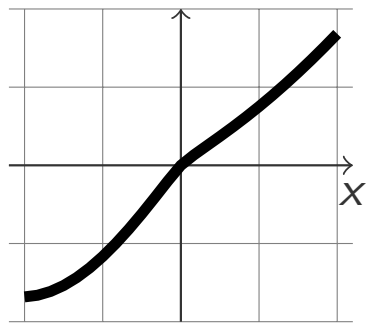




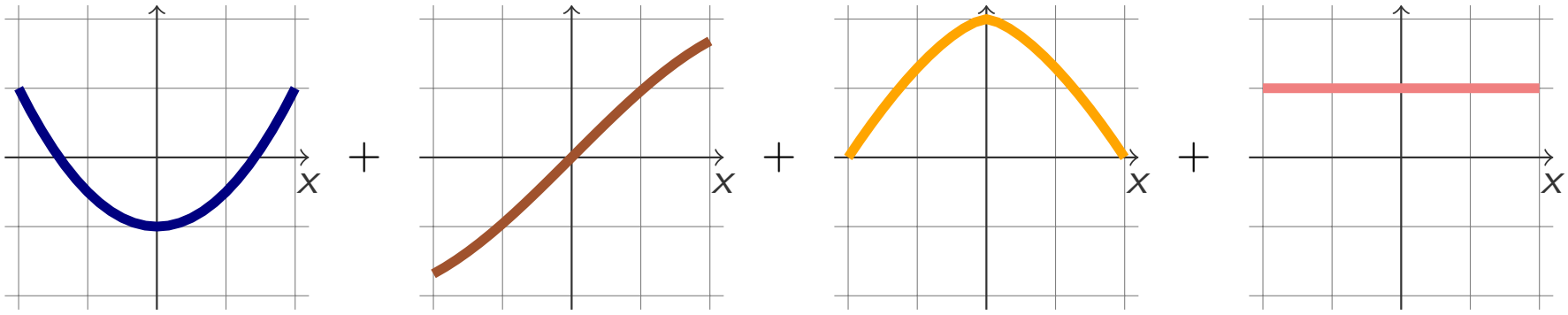
The Problem



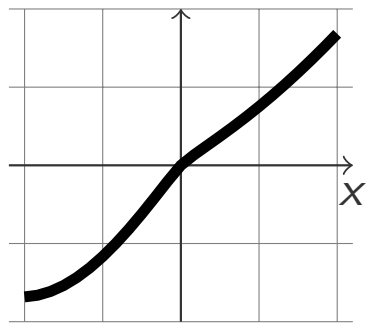
=



The Problem



=



No single agent knows the target function to optimize
The network does

Formally

$$\inf_{x \in X} \sum_{v \in V} f_v(x)$$

- $\mathcal{G} = (V, E)$ is the *graph* modelling the network
- f_v is the cost function of agent v
- X is a finite dimensional Euclidean space

Numerous works on that problem
Early work: Tsitsiklis '84 (all f_v equal)

An example in wireless sensor networks

Y_v = random observation of sensor v
x = unknown parameter to be estimated

Assume that

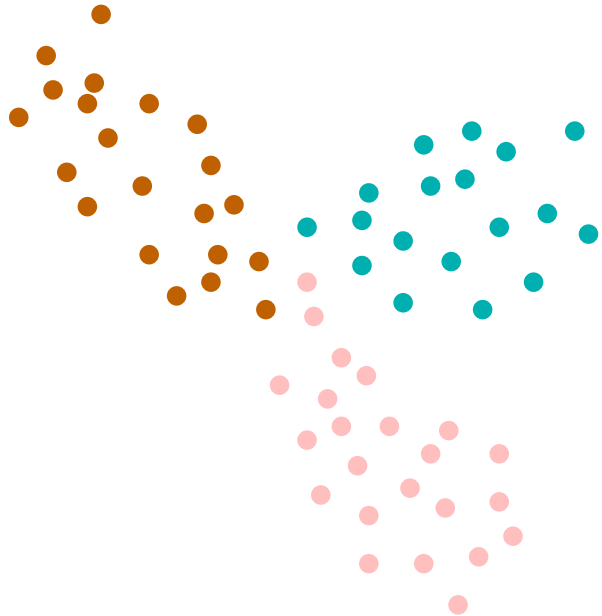
$$p(Y_1, \cdots, Y_N; x) = p_1(Y_1; x) \cdots p_N(Y_N; x)$$

The maximum likelihood estimate writes

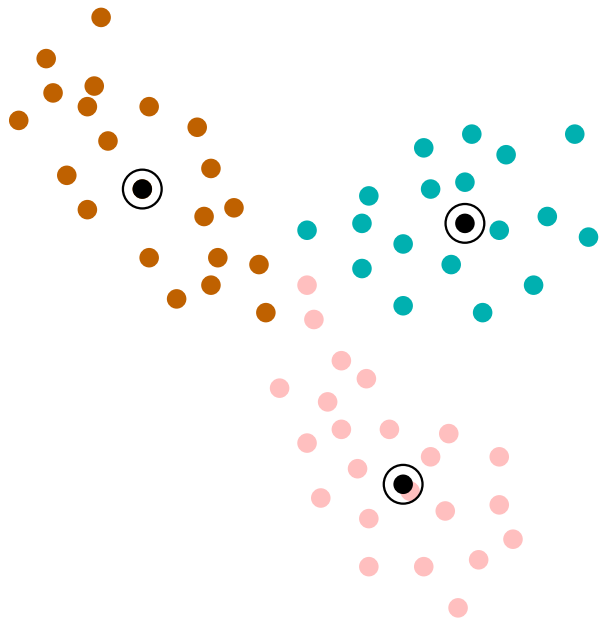
$$\hat{x} = \arg\max_x \sum_v \ln p_v(Y_v; x)$$

[Ribeiro et al.'06, Moura et al.'11]

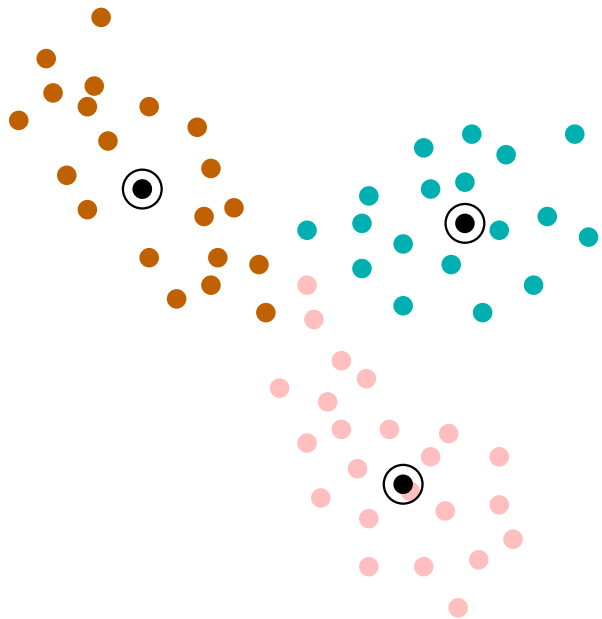
An example in machine learning



An example in machine learning



An example in machine learning

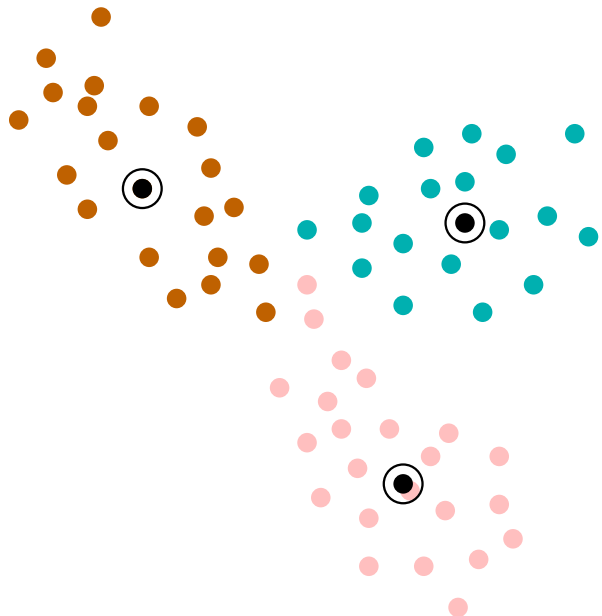


Centralized problem: For a data set $\{D_1, D_2, \dots\}$

$$\min_x \sum_{i=1,2\dots} \|D_i - q_x(D_i)\|^2$$

where $q_x(D)$ is the nearest point of D

An example in machine learning



Distributed problem: For N distributed data sets $\{D_{1,v}, D_{2,v}, \dots\}$ $(v \in V)$

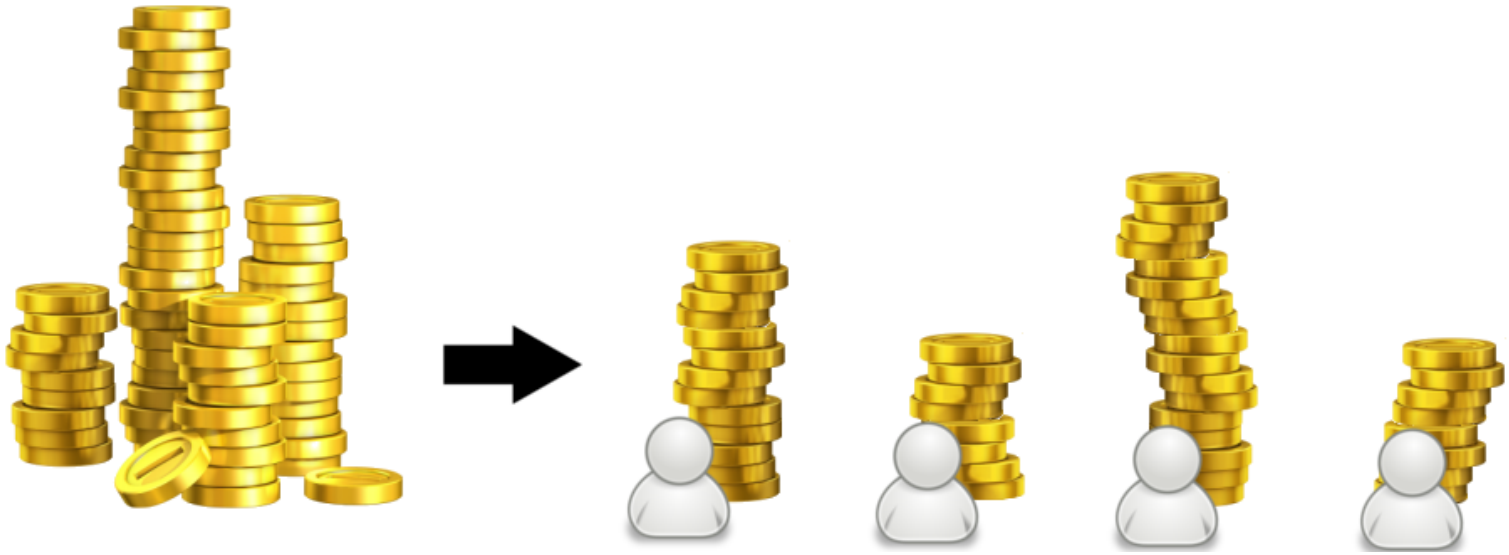
$$\min_x \sum_{v \in V} \left(\sum_{i=1,2\dots} \|D_{i,v} - q_x(D_{i,v})\|^2 \right)$$

where $q_x(D)$ is the nearest point of D

[Patra'11, Forero'11]

Outline

The sharing problem



Let $x(v)$ be the resource of an agent $v \in V$

- Agents share a resource b: $\sum_{v \in V} x(v) \leq b$
- Agent v gets reward $-f_v(x(v))$ for using resource $x(v)$
- Maximize the global reward

$$\inf_{x:\sum x(v) \leq b} \sum_{v \in V} f_v(x(v))$$

Equivalence between consensus and sharing

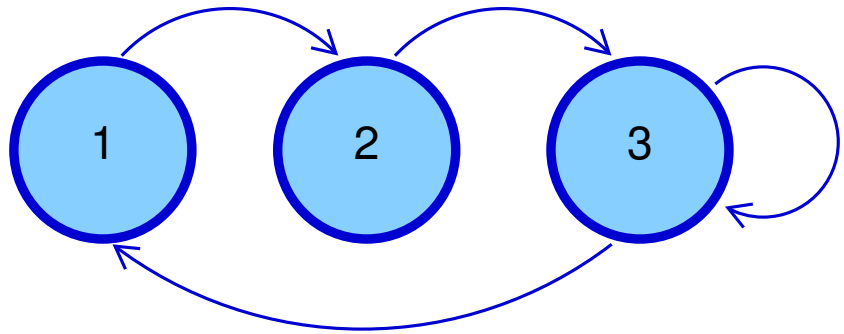
Claim:
The dual of a sharing problem is a consensus problem

Outline

Network model

A directed graph $\mathcal{G} = (V, E)$ is formed by

- a finite set V of vertices
- a set $E \subset V \times V$ of directed edges



A *strongly connected* graph with one *self-loop*

An iterative algorithm is said **distributed on the graph** if, at any iteration:

Agent v can receive information from w only if $(v, w) \in E$

The average consensus problem

Average consensus problem

Given an initial value $x_0(v) \in \mathbb{R}$ of each agent v, compute distributively

$$\overline{x}_0 \triangleq \frac{1}{N} \sum_{v \in V} x_0(v)$$

- Very special case of optimization problem! Just set $f_v(x) = (x - x_0(v))^2$
- Useful to adress more general optimization problems

The agreement algorithm (De Groot'74)

Algorithm:
Each agent maintains an estimate $x_n(v)$. The update is:

$$\forall v \in V, \quad x_{n+1}(v) = \sum_{w \in V} A(v,w) x_n(w)$$

Assumptions:

- $A(v,w) \geq 0$ are non-negative weights
- $A(v,w) > 0$ if and only if $(v,w) \in E$ (we say that A is *adapted* to $\mathcal{G}$)
- $\sum_w A(v,w) = 1$ for any $v \in V$

The agreement algorithm: Vector form

Agreement algorithm

$$\begin{aligned} x_n &= A\,x_{n-1} \\ &= A^n x_0 \end{aligned}$$

where $A = [A(u,v)]_{(u,v)\in V^2}$ is *non-negative*, *row-stochastic* and *adapted* to $\mathcal{G}$

Row stochasticity means:

$$A\mathbf{1} = \mathbf{1}$$

where $\mathbf{1} \triangleq \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

Discussion

What do we hope for?

$$\forall x_0,\ \lim_n x_n = \overline{x}_0 \mathbf{1} \quad (?)$$

- $x_n(w)$ cannot converge to $\overline{x}_0$ if no path from v to w ! (*e.g.* $A = I_N$)

  $\mathcal{G}$ must be connected
- A should preserve the average *i.e.* $\overline{x}_1 = \overline{x}_0$

  A must be doubly stochastic: $\mathbf{1}^* A = \mathbf{1}^*$
- Even then, convergence is not ensured. Set *e.g.*

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

A consequence of the Perron-Frobenius theorem

Definition: Matrix A is *primitive* if $A^m > 0$ for some $m \geq 1$

Property: If $\mathcal{G}$ is connected and has a self-loop, then A is primitive

Theorem

Let $A \geq 0$. The following statements are equivalent:

- For any x_0, $\lim_{n\to\infty} A^n x_0 = \overline{x}_0 \mathbf{1}$
- A is primitive and doubly stochastic

n.b. Many variants on that problem [Kempe et al.'03, Boyd et al.'06]

Outline

Outline

The setting

Consensus problem in optimization

$$\inf_{x \in X} f(x) \triangleq \sum_{v \in V} f_v(x)$$

Scenario

Centralized gradient algorithm

$$x_{n+1} = x_n - \gamma \nabla f(x_n)$$

Under some assumptions, achieves *linear convergence rate* in $\mathcal{O}(\beta^n)$, $(\beta < 1)$

Problem: ∇f is nowhere available

Distributed gradient algorithms: The Two Main Options

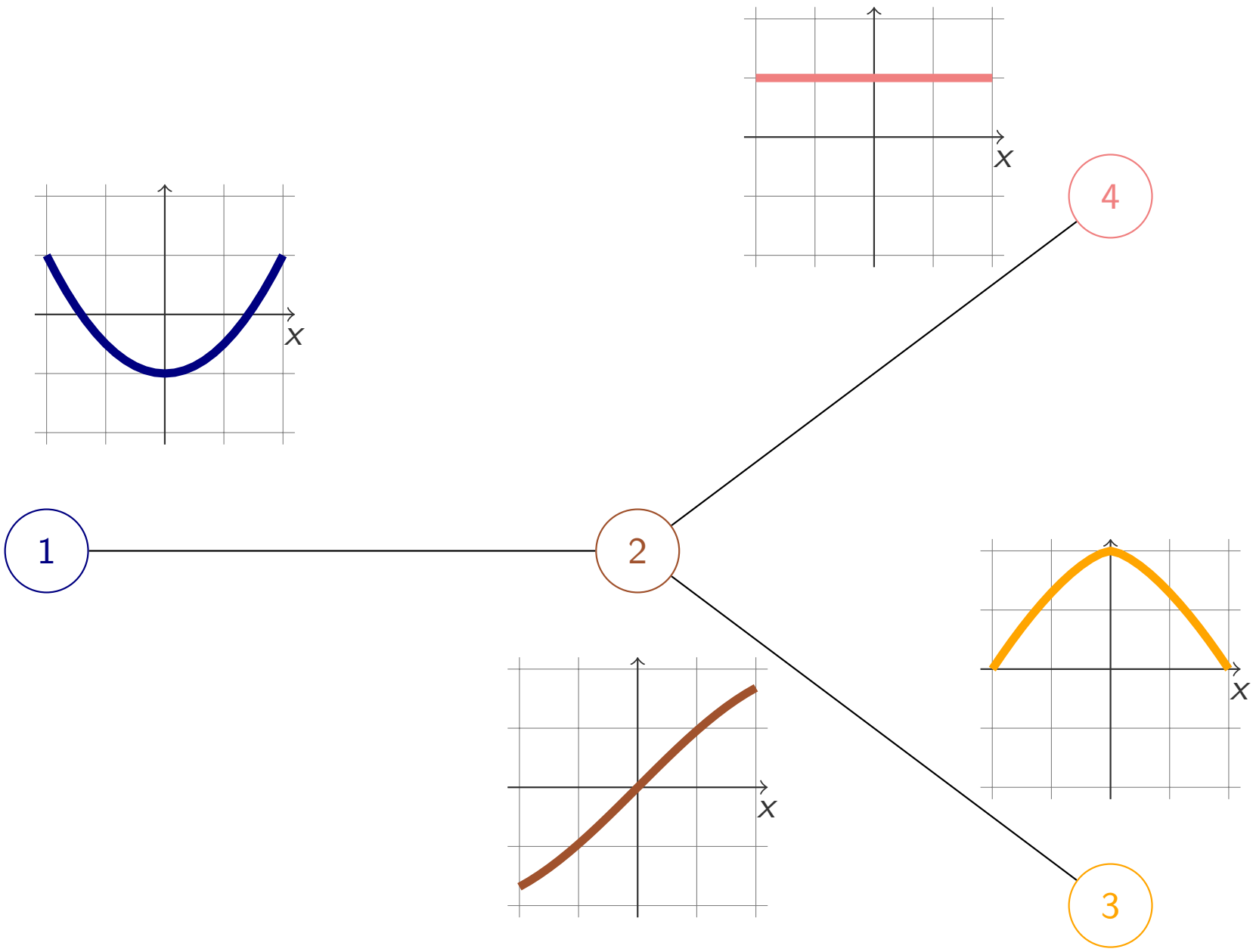
- Incremental

  [Widrow-Hoff'60], [Nedic-Bertsekas'01]

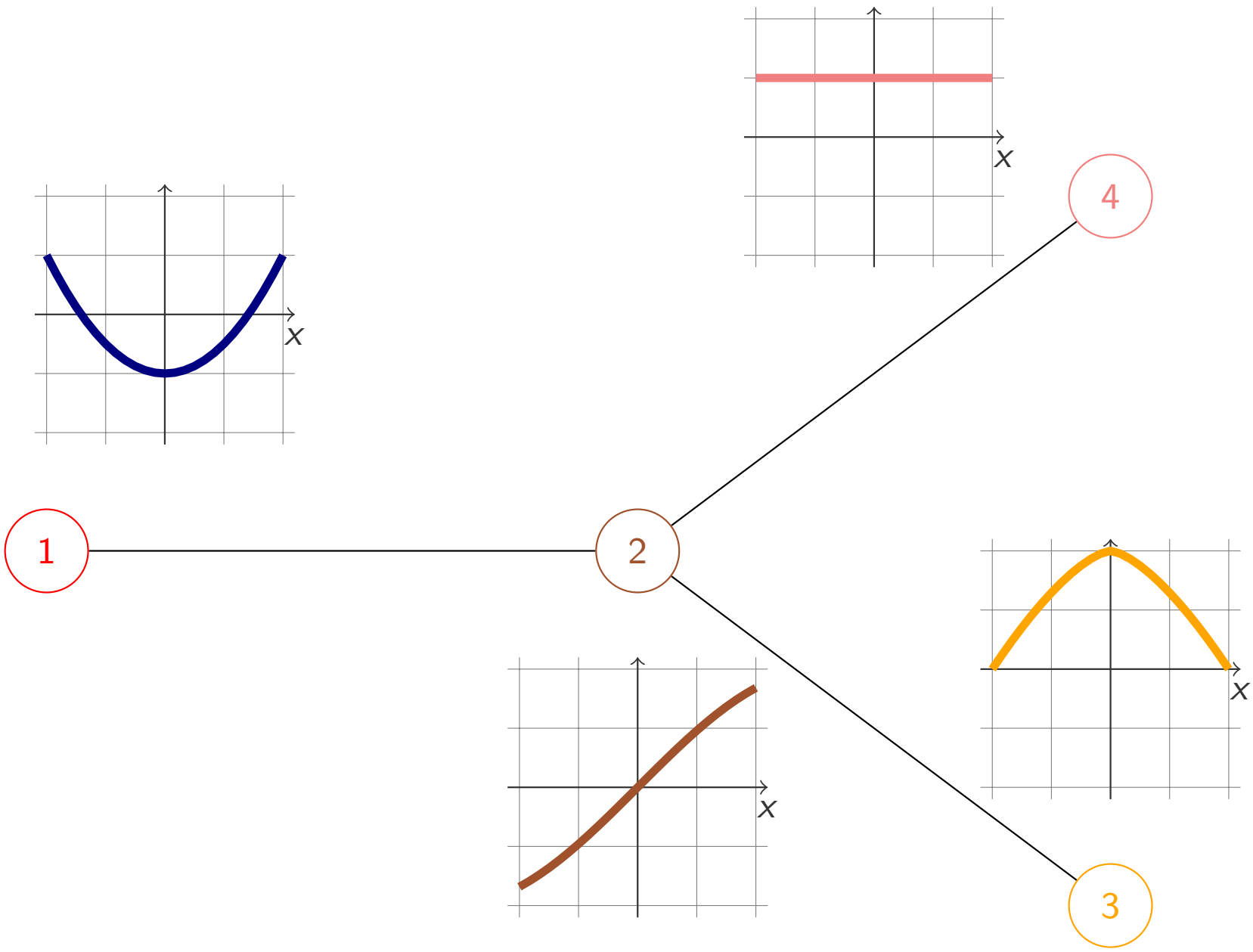
- Agreement

  [Tsitsiklis'84], [Kushner'87], [Sayed et al.'05], [Ram et al.'10], ...

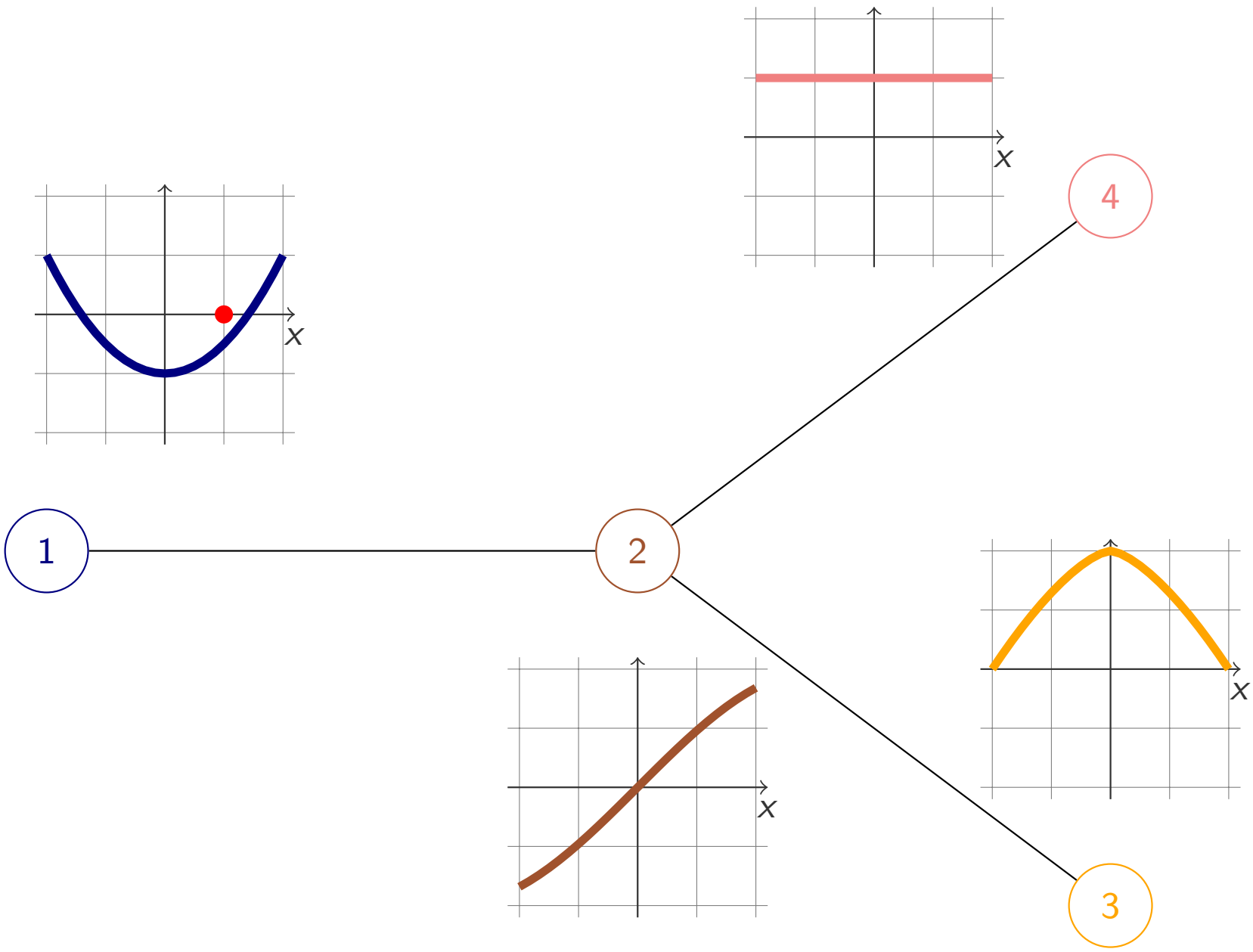
Incremental



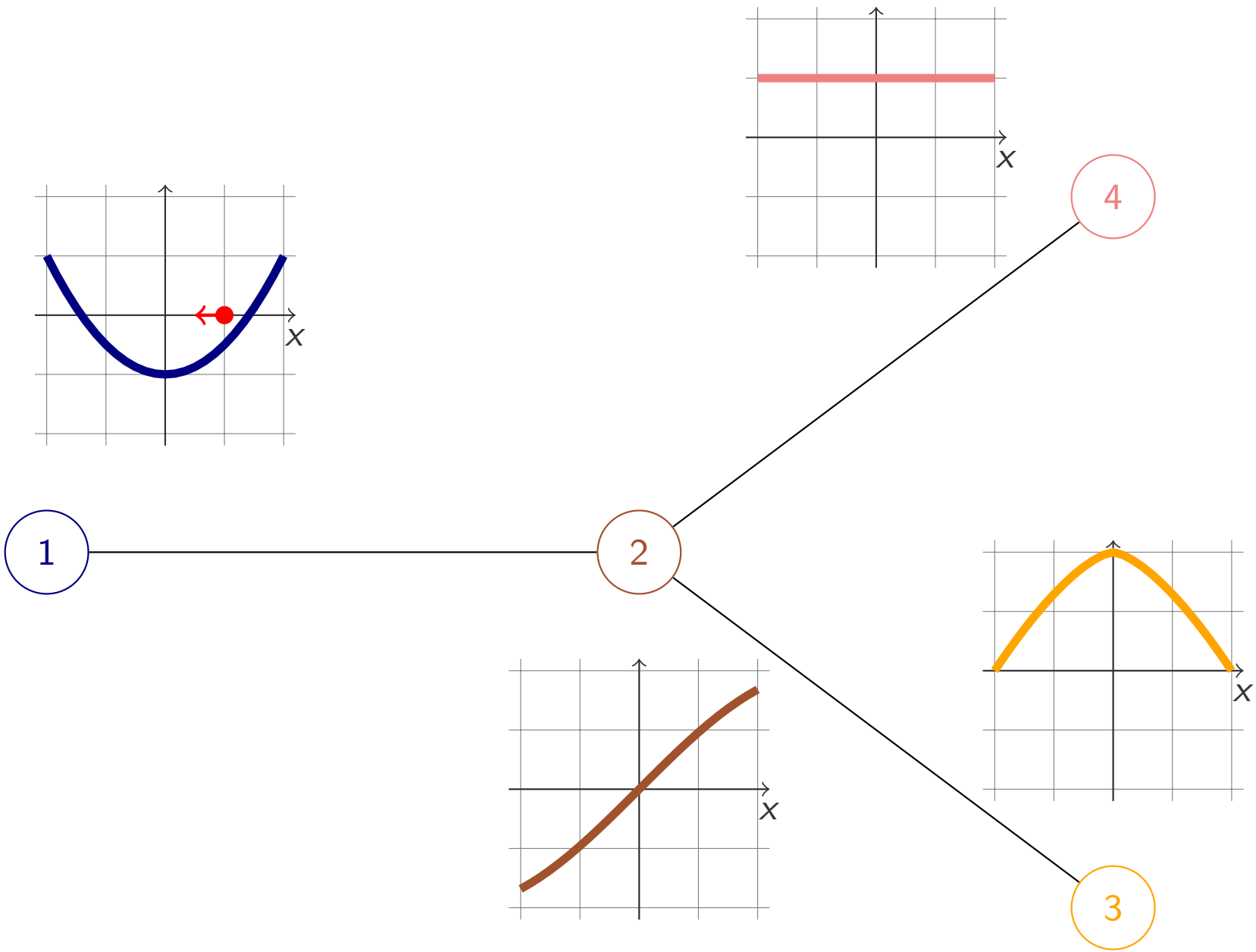
Incremental



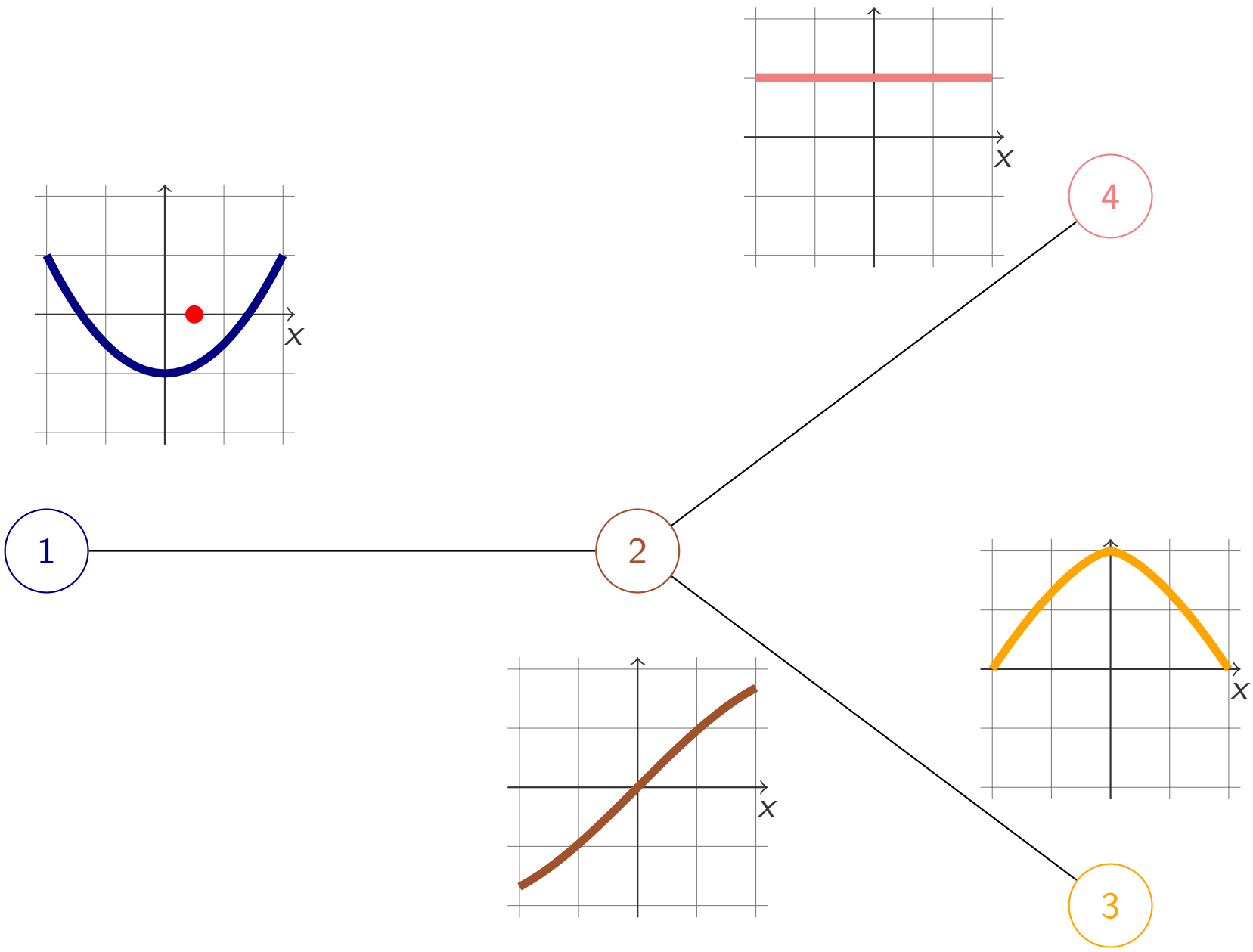
Incremental



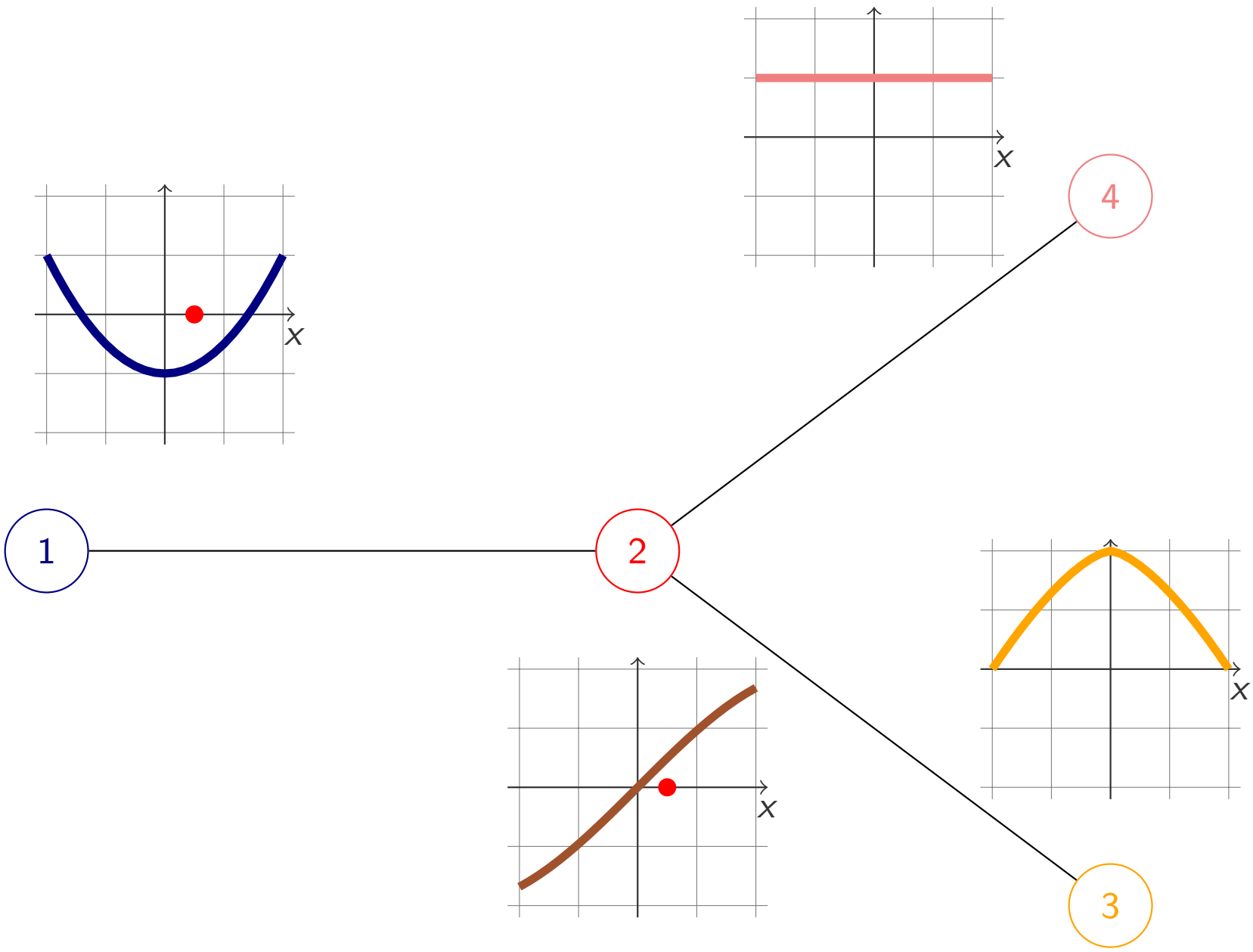
Incremental



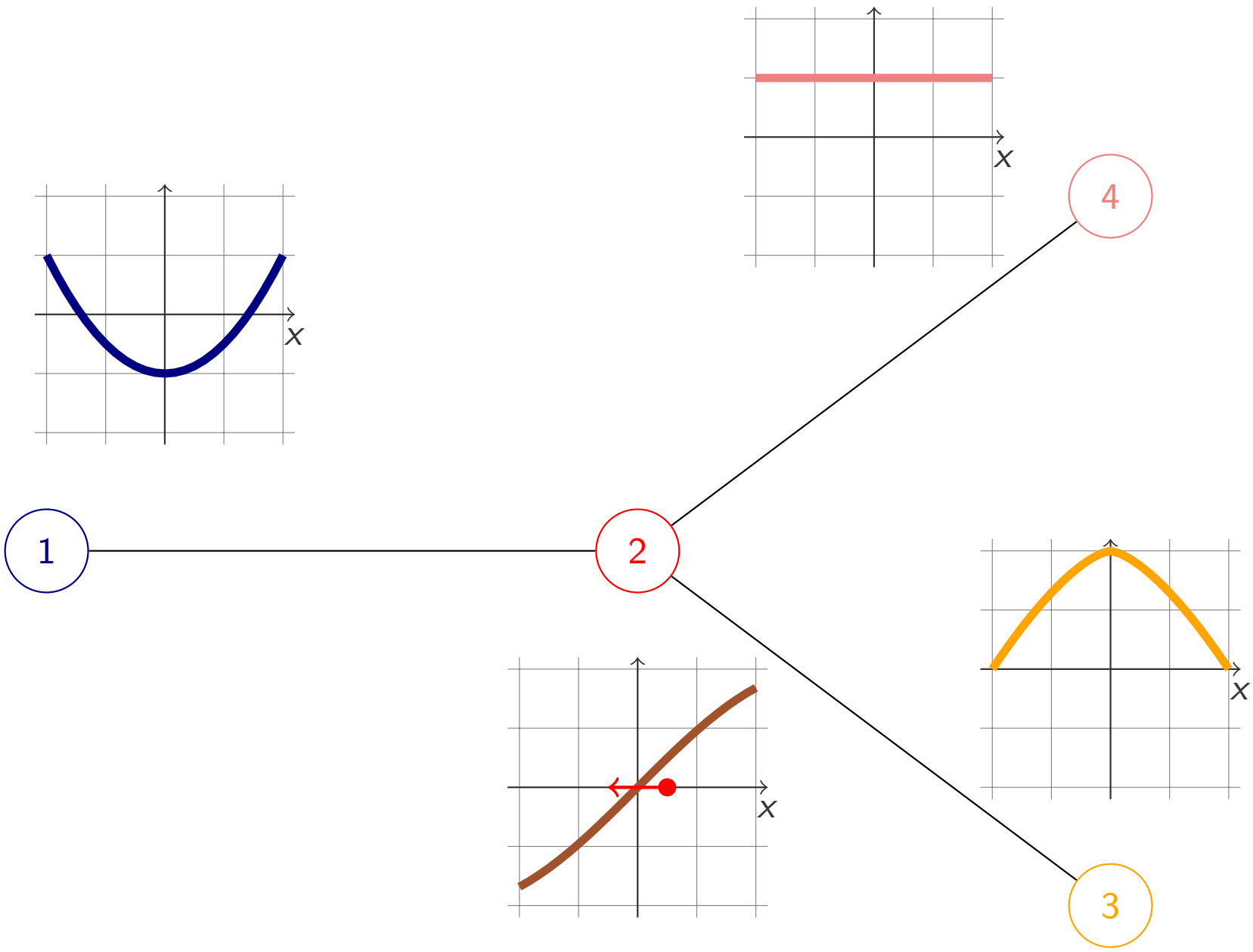
Incremental



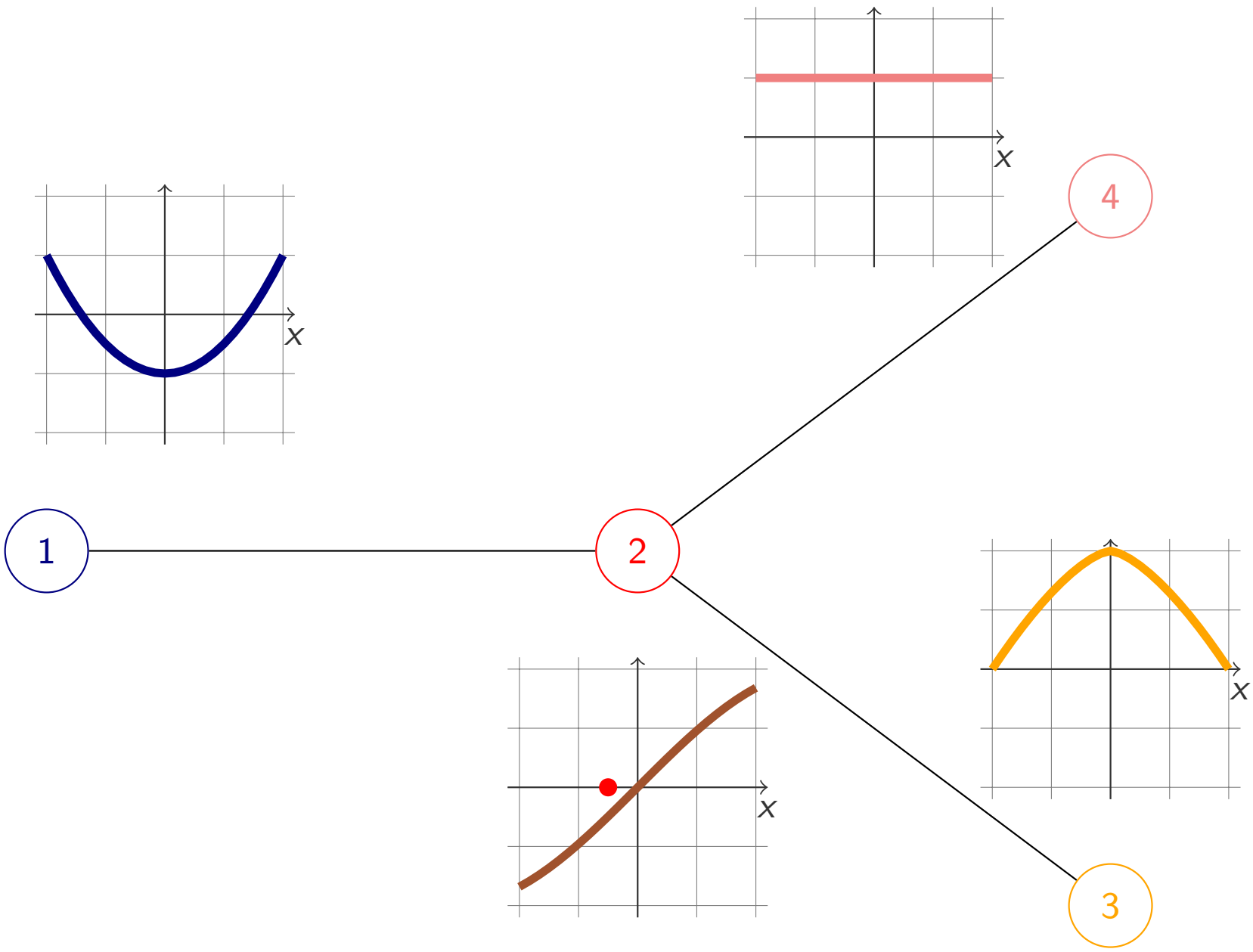
Incremental



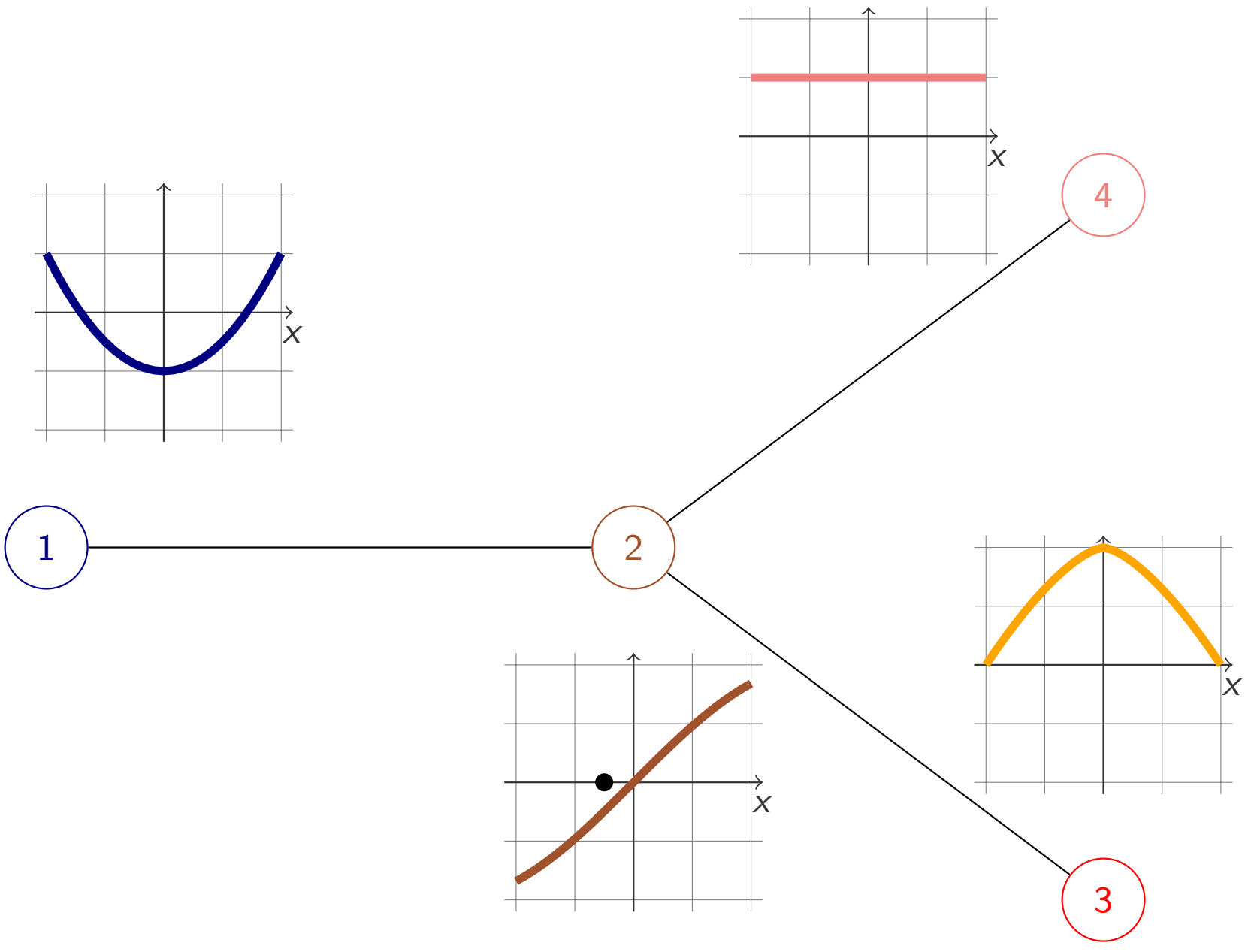
Incremental



Incremental



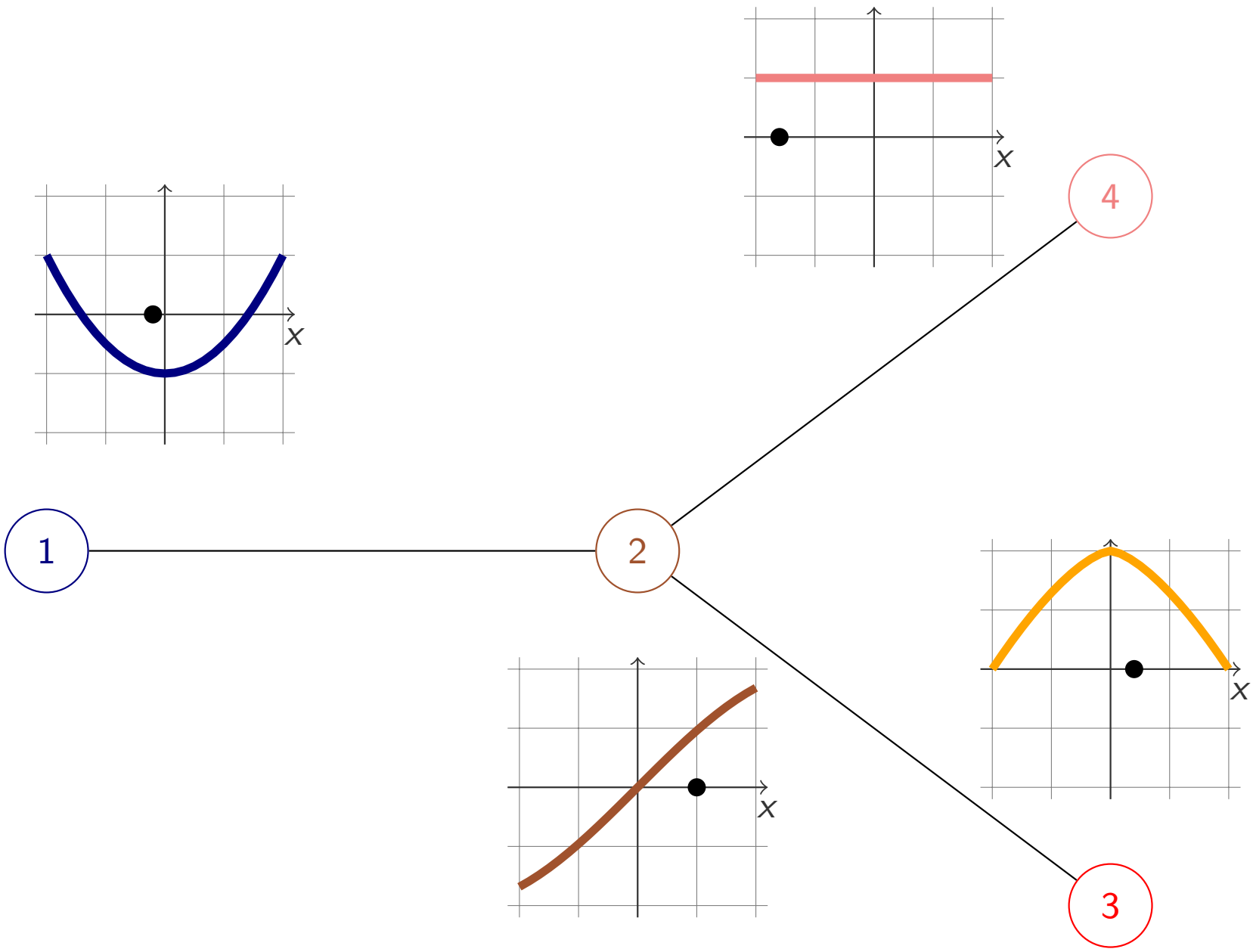
Incremental



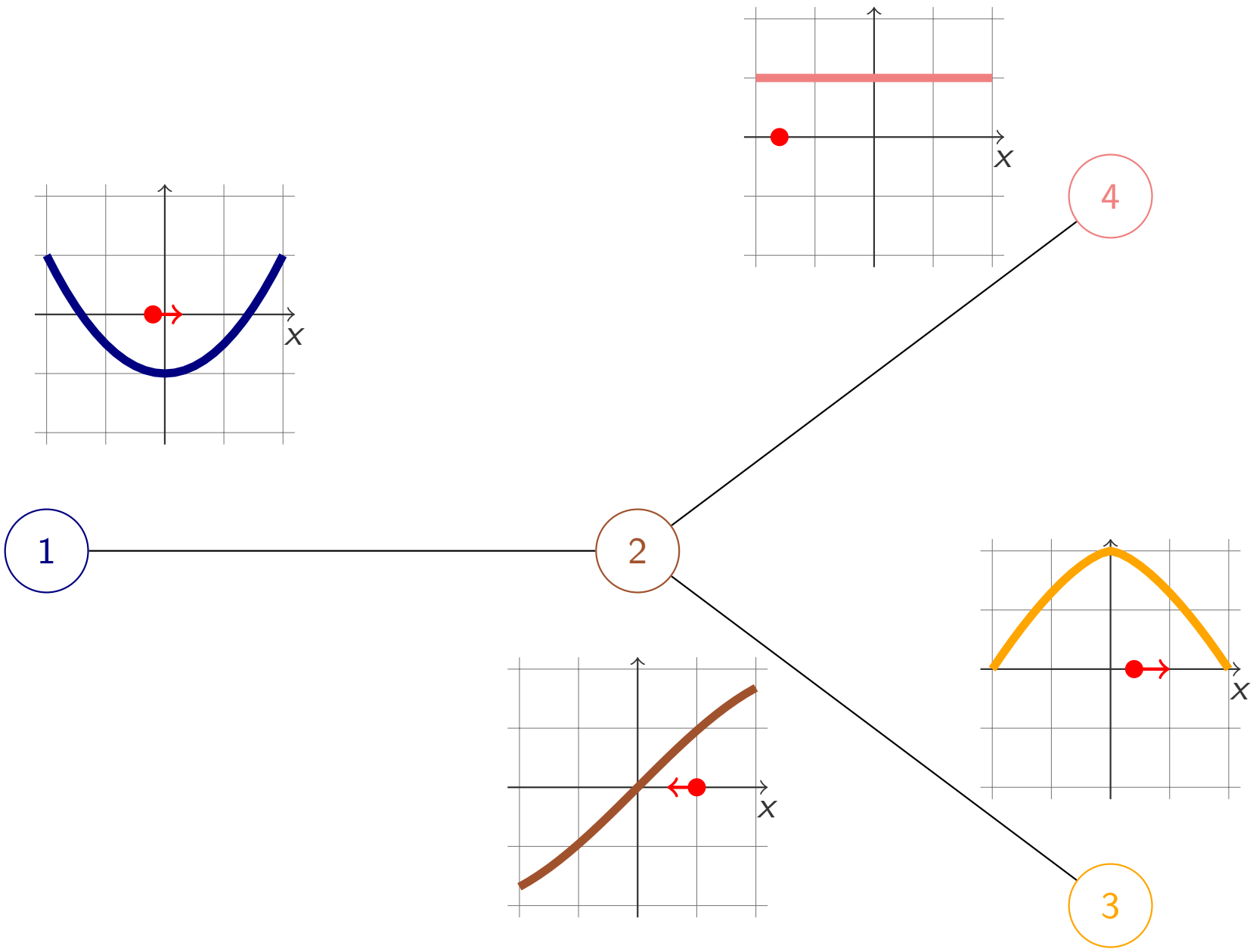
Agreement

Idea: couple gradient algorithm + agreement algorithm

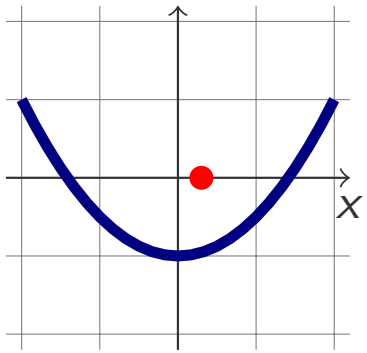
Agreement

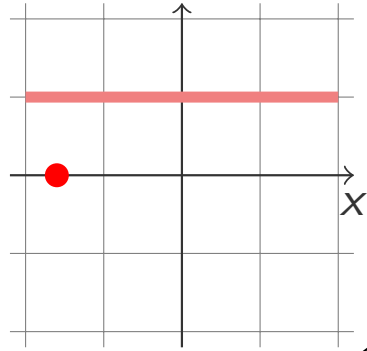


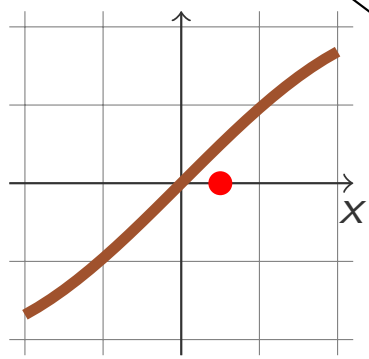
Agreement

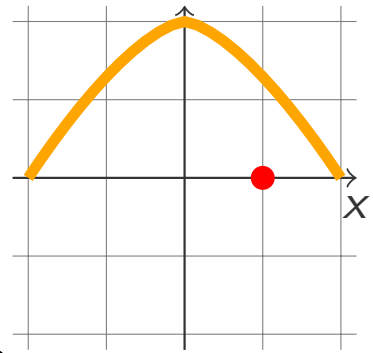


Agreement

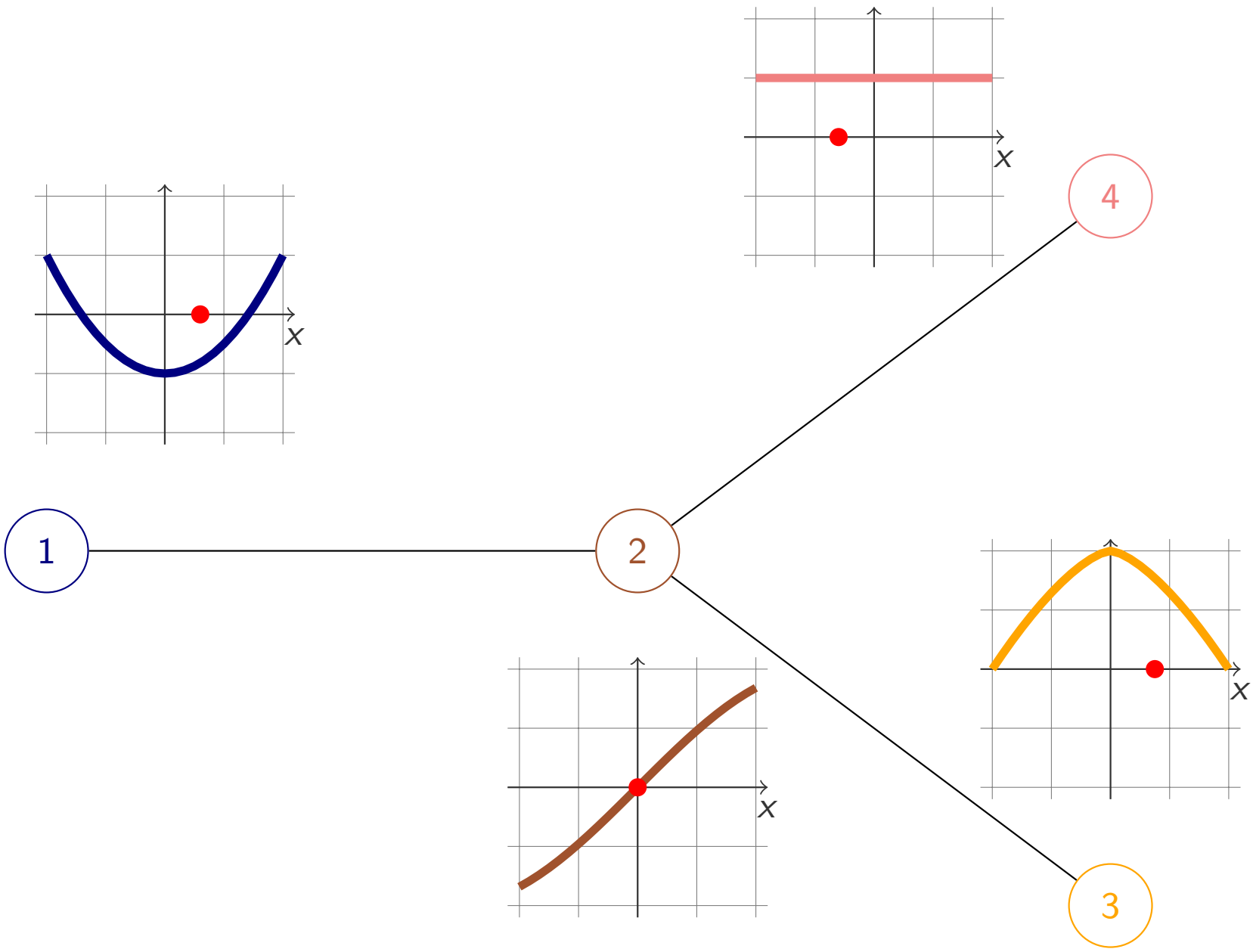








Agreement



Agreement, Formal

- [Local step] Each agent v generates a temporary update

$$\tilde{x}_{n+1}(v) = x_n(v) - \gamma_n \nabla f_v(x_n(v))$$

Agreement, Formal

- `[Local step]` Each agent v generates a temporary update

$$\tilde{x}_{n+1}(v) = x_n(v) - \gamma_n \nabla f_v(x_n(v))$$

- `[Agreement step]` Connected agents merge their temporary estimates

$$x_{n+1}(v) = \sum_{w=1}^{N} A(v,w)\, \tilde{x}_{n+1}(w)$$

Benefits & Drawbacks

Incremental

- Conceptually simple
- Needs Hamiltonian cycle (or at least a relaxed version)
- Concentrated information: less robust

Agreement

- No need for a Hamiltonian cycle
- Simple to implement

Benefits & Drawbacks

Incremental

- Conceptually simple
- Needs Hamiltonian cycle (or at least a relaxed version)
- Concentrated information: less robust

Agreement

- No need for a Hamiltonian cycle
- Simple to implement

Outline

Distributed algorithm: Vector notation

Let $X = \mathbb{R}$ for simplicity. Recall notation

$$F(x) \triangleq \sum_{v \in V} f_v(x(v))$$

- `[Local step]`
$$\tilde{x}_{n+1} = x_n - \gamma_n \nabla F(x_n)$$
- `[Agreement step]`
$$x_{n+1} = A\,\tilde{x}_{n+1}$$

$$x_{n+1} = A\,(x_n - \gamma_n \nabla F(x_n))$$

Assumption

Except in special cases, convergence to the sought minimizers fails unless:

Assumption

- A is doubly stochastic
- A is primitive

Agreement

Assume that $C := \limsup_n \|\nabla F(x_n)\|$ is finite. Define

$$J = \frac{\mathbf{1}\mathbf{1}^*}{N} \qquad J_\perp = I_N - J$$

Compute the *disagreement vector*:

$$J_\perp x_{n+1} \quad = \quad J_\perp A\left(x_n - \gamma_n \nabla F(x_n)\right)$$

Agreement

Assume that $C := \limsup_n \|\nabla F(x_n)\|$ is finite. Define

$$J = \frac{\mathbf{1}\mathbf{1}^*}{N} \qquad J_\perp = I_N - J$$

Compute the *disagreement vector*:

$$J_\perp x_{n+1} \quad = \quad J_\perp A J_\perp \left(x_n - \gamma_n \nabla F(x_n)\right)$$

Agreement

Assume that $C := \limsup_n \|\nabla F(x_n)\|$ is finite. Define

$$J = \frac{\mathbf{1}\mathbf{1}^*}{N} \qquad J_\perp = I_N - J$$

Compute the *disagreement vector*:

$$J_\perp x_{n+1} \quad = \quad J_\perp A J_\perp \left(J_\perp x_n - \gamma_n \nabla F(x_n)\right)$$

Agreement

Assume that $C := \limsup_n \|\nabla F(x_n)\|$ is finite. Define

$$J = \frac{\mathbf{1}\mathbf{1}^*}{N} \qquad J_\perp = I_N - J$$

Compute the *disagreement vector*:

$$J_\perp x_{n+1} \quad = \quad J_\perp A J_\perp \left(J_\perp x_n - \gamma_n \nabla F(x_n)\right)$$

Denote by σ the spectral norm of $J_\perp A J_\perp$. We have $\sigma < 1$.

$$\|J_\perp x_{n+1}\| \quad \leq \quad \sigma \left(\|J_\perp x_n\| + \gamma_n \|\nabla F(x_n)\|\right)$$

Disagreement vector

Assume $\gamma_n/\gamma_{n+1} \to 1$.

$$\limsup_n \frac{\|J_\perp x_n\|}{\gamma_n} \leq \frac{\sigma C}{1-\sigma}$$

Remarks

1. In order that $\|J_\perp x_n\| \to 0$, vanishing step size is needed

$$\gamma_n \to 0$$

   except if *e.g.* all f_v's have a common minimizer ($C = 0$)

2. The disagreement tends to zero at rate γ_n

$$\|J_\perp x_n\| = \mathcal{O}(\gamma_n)$$

3. Factor $\frac{\sigma}{1-\sigma}$ quantifies the network effect [Duchi et al.'11]

Convergence of the network average

It remains to study the network-average

$$\overline{x}_n = \frac{\mathbf{1}^* x_n}{N}$$

As $\mathbf{1}^* A = \mathbf{1}^*$,

$$\overline{x}_{n+1} \quad = \quad \overline{x}_n - \frac{\gamma_n}{N} \mathbf{1}^* \nabla F(x_n)$$

Convergence of the network average

It remains to study the network-average

$$\overline{x}_n = \frac{\mathbf{1}^* x_n}{N}$$

As $\mathbf{1}^* A = \mathbf{1}^*$,

$$\begin{aligned} \overline{x}_{n+1} &= \overline{x}_n - \frac{\gamma_n}{N} \mathbf{1}^* \nabla F(x_n) \\ &\simeq \overline{x}_n - \frac{\gamma_n}{N} \mathbf{1}^* \nabla F(\overline{x}_n \mathbf{1}) \end{aligned}$$

Convergence of the network average

It remains to study the network-average

$$\overline{x}_n = \frac{\mathbf{1}^* x_n}{N}$$

As $\mathbf{1}^* A = \mathbf{1}^*$,

$$\begin{aligned} \overline{x}_{n+1} &= \overline{x}_n - \frac{\gamma_n}{N} \mathbf{1}^* \nabla F(x_n) \\ &\simeq \overline{x}_n - \frac{\gamma_n}{N} \nabla f(\overline{x}_n) \end{aligned}$$

The network average nearly behaves as a gradient descent on f.

Convergence result

Assumptions

$$\sum_n \gamma_n = +\infty, \quad \sum_n \gamma_n^3 < \infty$$

Moreover, assume that

- ∇f_v is lispchitz continuous for all v
- $f \triangleq \sum_v f_v$ is coercive and $\{\nabla f = 0\}$ is locally finite

Convergence

There exists $x^\star \in \{\nabla f = 0\}$ such that

$$\lim_{n \to \infty} \overline{x}_n = x^\star$$

Asymptotic rate of convergence

Assumptions

- $\nabla^2 f(x^\star) \succ 0$
- $\gamma_n \propto 1/n^\alpha$ for $0 < \alpha \leq 1$

Then, optimal convergence rate is achieved for $\gamma_n \propto \frac{1}{n}$ and

Convergence rate (smooth case)

$$x_n = x^\star \mathbf{1} + \mathcal{O}\left(\frac{\log n}{n}\right)$$

Quite far from the *linear convergence rate* $\mathcal{O}(\beta^n)$ of the centralized case

Outline

Algorithm

Assumptions

- All f_v convex non-negative
- f_v are L-lipschitz
- $f = \sum_v f_v$ achieves its minimum at $x^\star$

Distributed subgradient algorithm

$$x_{n+1} = A(x_n - \gamma_n g_n)$$

where for any $v \in V$

$$g_n(v) \in \partial f_v(x_n(v))$$

Convergence result (1/2)

Define the time-averaged estimate for all $v \in V$

$$\hat{x}_n(v) = \frac{\sum_{k \leq n} \gamma_k x_k(v)}{\sum_{k \leq n} \gamma_k}$$

Error bound (Nedic, Ozdaglar'10)

$$f(\hat{x}_n(v)) - f(x^\star) \leq \frac{\frac{1}{2}\|\overline{x}_0 - x^\star\|^2 + (1 + NET)L^2 \sum_{k \leq n} \gamma_k^2}{\sum_{k \leq n} \gamma_k}$$

where NET grasps the excess-bound due to the distributed setting

$$NET = \frac{\sigma}{1 - \sigma}\left(\sqrt{N} + \frac{1}{\sqrt{N}}\right)$$

Convergence result (2/2)

- The bound is exact (*i.e.* non-asymptotic)
- Set $\gamma_n \propto \frac{1}{\sqrt{n}}$

$$\text{The bound is } \mathcal{O}\left(\frac{\log n}{\sqrt{n}}\right)$$

- $\log n$ factor can be saved following [Nesterov'05]
  [Duchi et al.'11] couples Nesterov algorithm + agreement algorithm

Optimal rate of the centralized case

Outline

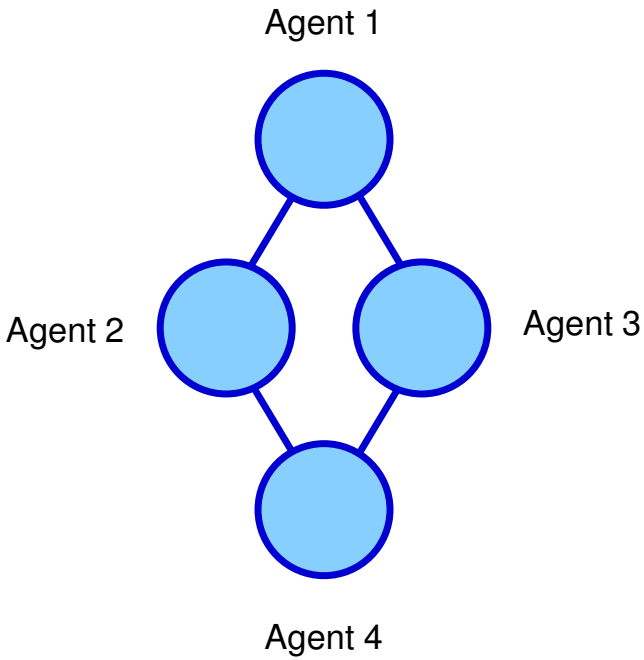
More problems

1. **Asynchronism**
   Some agents are active at time n, others aren't

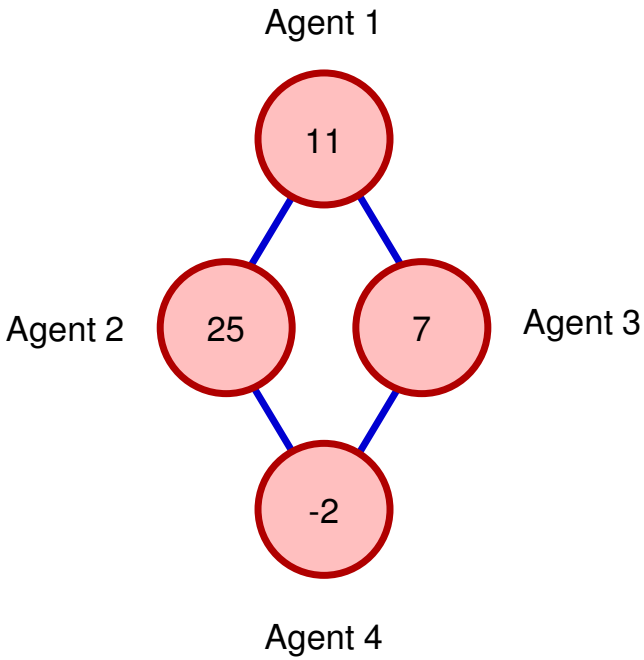
2. **Noise**
   Gradients may be observed up to a random noise (online algorithms)

3. **Constraints**
   $$\text{Minimize } \sum_{v \in V} f_v(x) \text{ subject to } x \in G$$
   where G is a a closed convex set

An asynchronous agreement protocol (Aysal et al.'09)



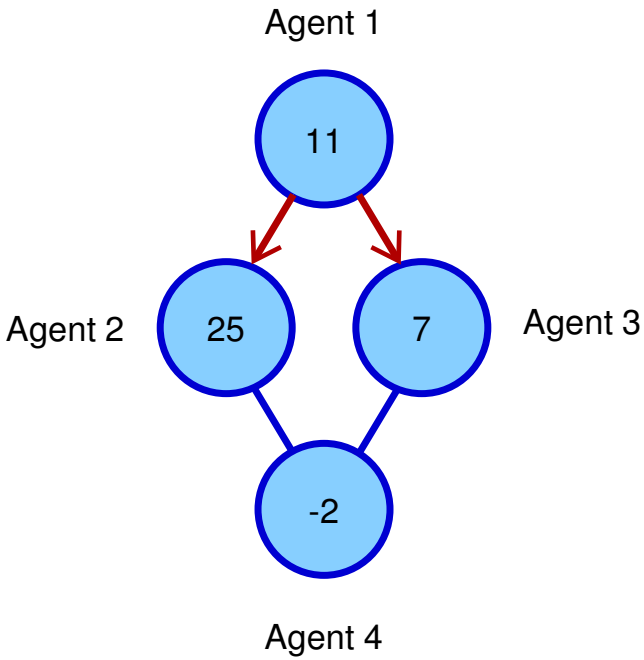
An asynchronous agreement protocol (Aysal et al.'09)



[Local step]

$$\tilde{x}_{n+1} = x_n - \gamma_n \nabla F(x_n)$$

An asynchronous agreement protocol (Aysal et al.'09)

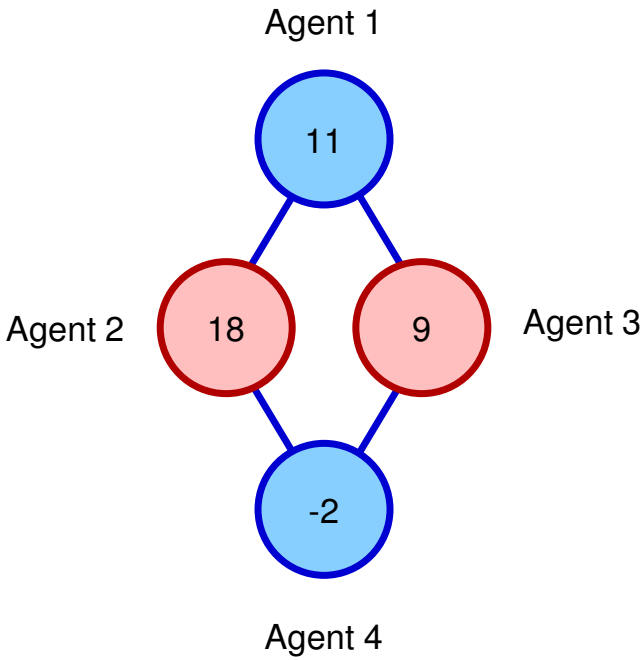


[`Local step`]

$$\tilde{x}_{n+1} = x_n - \gamma_n \nabla F(x_n)$$

[Agreement step]

An asynchronous agreement protocol (Aysal et al.'09)



[Local step]

$$\tilde{x}_{n+1} = x_n - \gamma_n \nabla F(x_n)$$

[Agreement step]

$$x_{n+1} = A_{n+1}\tilde{x}_{n+1}$$

$$A_{n+1} = \begin{pmatrix} 1 & & & \\ 0.5 & 0.5 & & \\ 0.5 & & 0.5 & \\ & & & 1 \end{pmatrix}$$

- row-stochastic but **not** column-stochastic $\mathbf{1}^* A_n \neq \mathbf{1}^*$
- hopefully, column stochasticity is satisfied *in average* $\mathbf{1}^* \mathbb{E}(A_n) = \mathbf{1}^*$

Distributed Robbins-Monro algorithm

Our problem

$$\inf_{x \in X} \sum_{v \in V} f_v(x)$$

Algorithm

$$x_{n+1} = A_{n+1} \left(x_n - \gamma_n \nabla F(x_n) + \gamma_n \xi_{n+1} \right)$$

where ξ_{n+1} is a martingale increment noise

$$\mathbb{E} \left(\xi_{n+1} \,|A_n, \xi_n, A_{n-1}, \xi_{n-1}, \cdots \right) = 0$$

Distributed Robbins-Monro algorithm

Our problem

$$\inf_{x \in X} \sum_{v \in V} f_v(x) \text{ subject to } x \in G$$

Algorithm

$$x_{n+1} = A_{n+1} \cdot \text{proj}_{G^{\otimes N}} \left[(x_n - \gamma_n \nabla F(x_n) + \gamma_n \xi_{n+1}) \right]$$

where ξ_{n+1} is a martingale increment noise

$$\mathbb{E}\left(\xi_{n+1} \,|A_n, \xi_n, A_{n-1}, \xi_{n-1}, \cdots\right) = 0$$

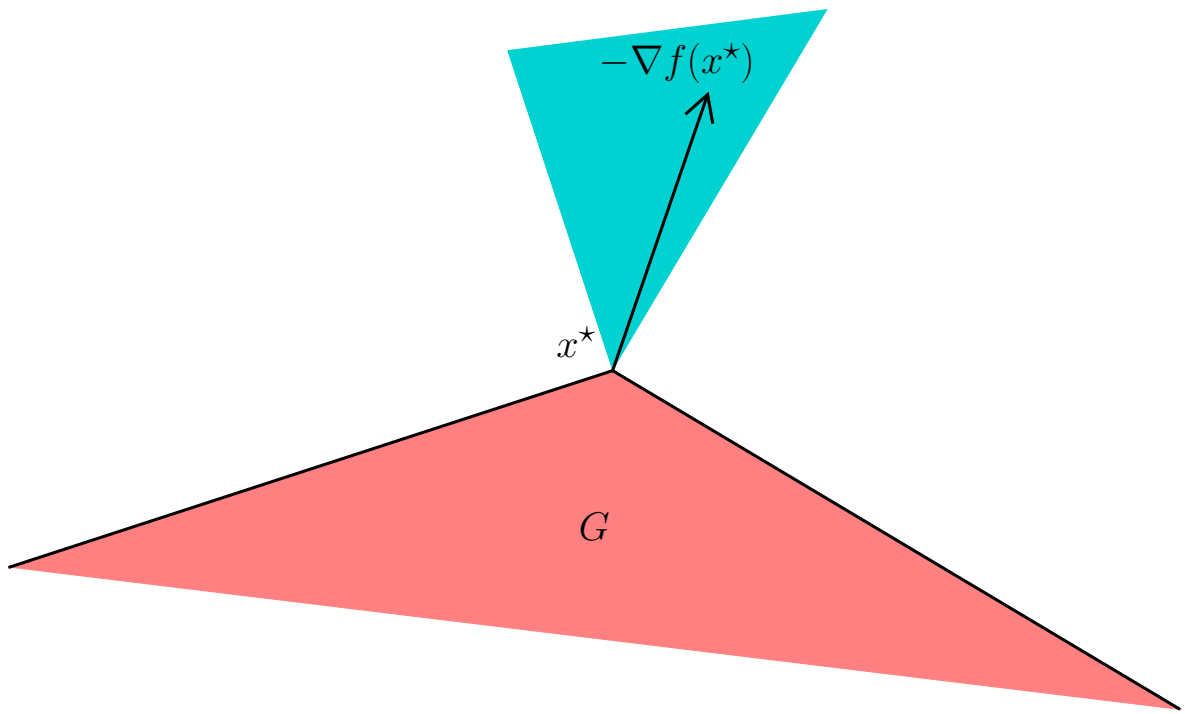
Consistency

Assume that $\mathbb{E}(A_n)$ is doubly stochastic and primitive

Theorem (Bianchi, Jakubowicz' 12)

Under suitable assumptions, x_n converges a.s. to $x^\star \mathbf{1}$ where

$$-\nabla f(x^\star) \in \mathcal{N}_G(x^\star)$$



One does not need A_n to be column-stochastic: broadcast protocol works!

Convergence rates

Assume that $x^\star$ lies in the interior of G and $\nabla^2 f(x^\star) \succ 0$

Theorem (Morral et al.'12)

Under suitable assumptions

- $J_\perp x_n = \mathcal{O}_P(\gamma_n)$
- For all $v \in V$, $\sqrt{\gamma_n}^{-1}(\overline{x}_n - x^\star) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Sigma_{OPT} + \Sigma_{NET})$

Conclusions

- Convergence rate $\sqrt{\gamma_n}$ is identical to the centralized case
  Optimal rate $1/\sqrt{n}$ achieved when $\gamma_n = 1/n$
- However, an excess-variance Σ_{NET} occurs
- $\Sigma_{NET} = 0$ if A_n is doubly-stochastic: same performance as centralized!

Σ_{NET} quantifies the price to pay for using uncoordinated weights

Outline

Outline

Consensus problem reformulated

All functions $f_v : X \to \mathbb{R}$ are assumed convex. Consider the problem:

$$\inf_{x \in X} \sum_{v \in V} f_v(x)$$

Set $F(x) = \sum_v f_v(x(v))$. The problem is equivalent to

$$\inf_{x \in X^N} F(x) + \iota_{\mathrm{sp}(\mathbf{1})}(x)$$

$$\text{where } \iota_{\mathrm{sp}(\mathbf{1})}(x) \quad = \quad \begin{cases} 0 & \text{if } x(1) = \cdots = x(N) \\ +\infty & \text{otherwise} \end{cases}$$

- F is separable in $x(1), \ldots, x(N)$
- $\iota_{\mathrm{sp}(\mathbf{1})}$ couples the variables but is simple

Alternating Direction Method of Multipliers (ADMM)

Define for any proper closed convex function h

$$\text{prox}_h(x) = \arg\min_y \; h(y) + \frac{1}{2}\|y - x\|^2$$

Algorithm: Set $\rho > 0$.

$$\begin{aligned} x_{n+1} &= \text{prox}_{\frac{1}{\rho}F}(z_n - \frac{\lambda_n}{\rho}) \\ z_{n+1} &= \text{prox}_{\frac{1}{\rho}\iota_{\text{sp}(\mathbf{1})}}(x_{n+1} + \frac{\lambda_n}{\rho}) \\ \lambda_{n+1} &= \lambda_n + \rho(x_{n+1} - z_{n+1}) \end{aligned}$$

- λ_n converges to a solution to the dual problem $\min_\lambda F^\star(-\lambda) + \iota^\star_{\text{sp}(\mathbf{1})}(\lambda)$
- x_n converges to a solution to the primal problem

Alternating Direction Method of Multipliers (ADMM)

Define for any proper closed convex function h

$$\text{prox}_h(x) = \arg\min_y \; h(y) + \frac{1}{2}\|y - x\|^2$$

Algorithm: Set $\rho > 0$.

$$\begin{aligned} x_{n+1} &= \text{prox}_{\frac{1}{\rho}F}(z_n - \frac{\lambda_n}{\rho}) \quad \rightarrow \text{ separable} \\ z_{n+1} &= \text{prox}_{\frac{1}{\rho}\iota_{\text{sp}(\mathbf{1})}}(x_{n+1} + \frac{\lambda_n}{\rho}) \quad \rightarrow \text{ projection} \\ \lambda_{n+1} &= \lambda_n + \rho(x_{n+1} - z_{n+1}) \end{aligned}$$

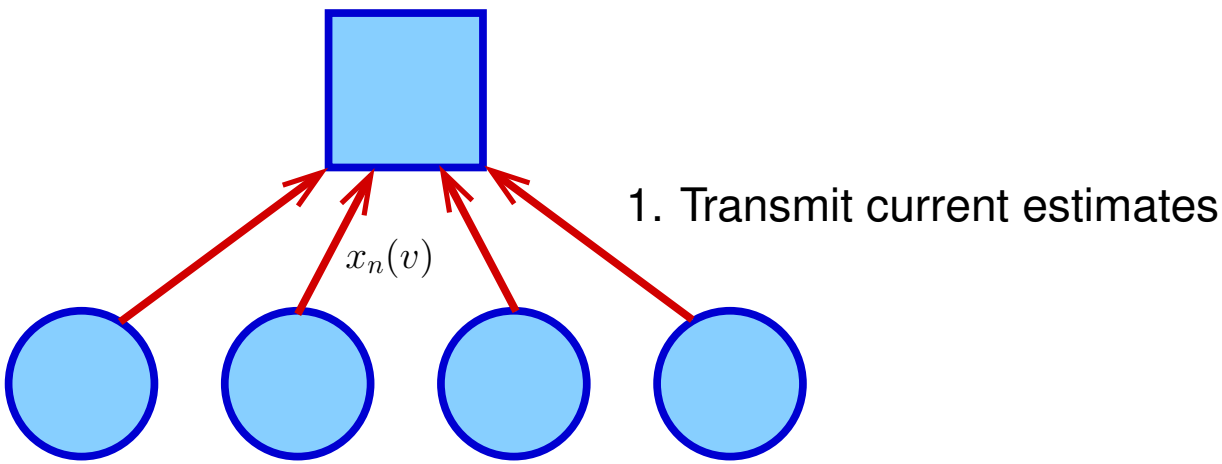
- λ_n converges to a solution to the dual problem $\min_\lambda F^\star(-\lambda) + \iota^\star_{\text{sp}(\mathbf{1})}(\lambda)$
- x_n converges to a solution to the primal problem

ADMM illustrated

Set $\beta_n = \lambda_n/\rho$

Algorithm (see e.g. [Boyd'11])

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \overline{x}_n \\ x_{n+1}(v) &= \text{prox}_{\frac{1}{\rho} f_v}\left(\overline{x}_n - \beta_n(v)\right) \end{aligned}$$

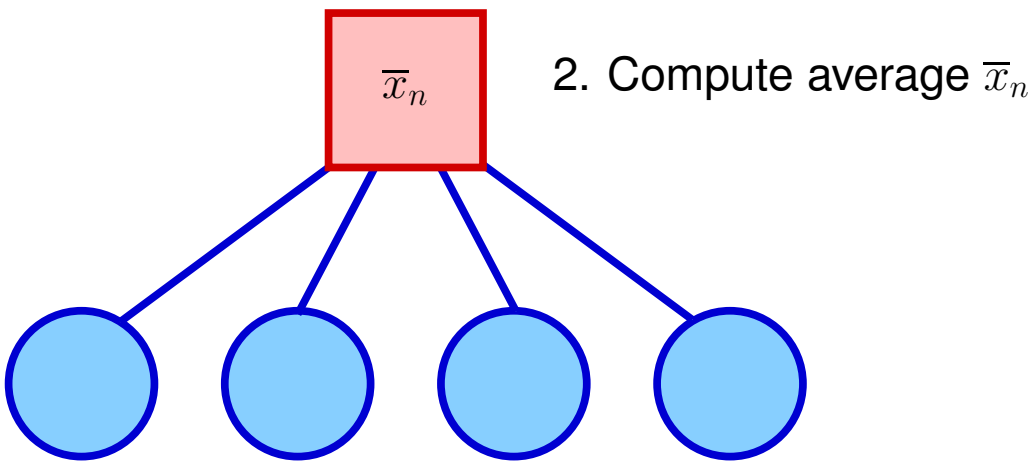


ADMM illustrated

Set $\beta_n = \lambda_n/\rho$

Algorithm (see e.g. [Boyd'11])

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \overline{x}_n \\ x_{n+1}(v) &= \text{prox}_{\frac{1}{\rho} f_v}\left(\overline{x}_n - \beta_n(v)\right) \end{aligned}$$

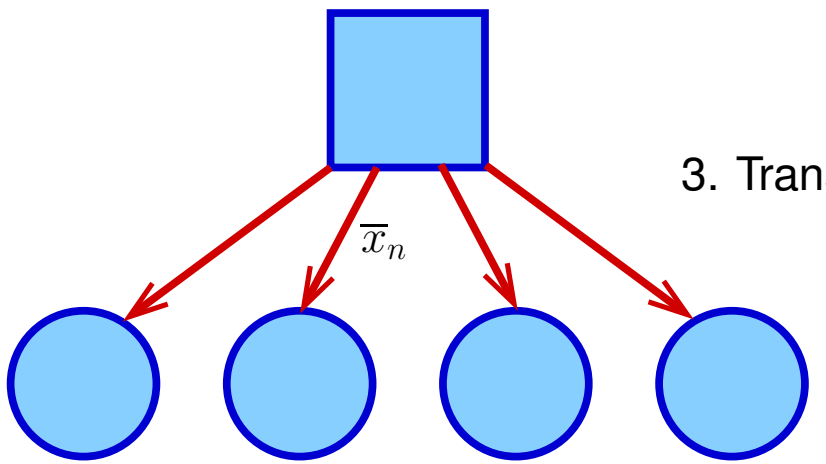


ADMM illustrated

Set $\beta_n = \lambda_n/\rho$

Algorithm (see e.g. [Boyd'11])

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \overline{x}_n \\ x_{n+1}(v) &= \text{prox}_{\frac{1}{\rho} f_v}\left(\overline{x}_n - \beta_n(v)\right) \end{aligned}$$



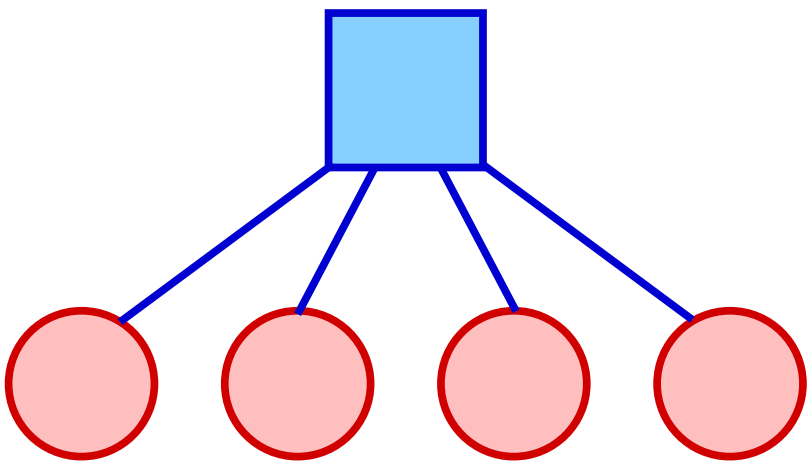
3. Transmit $\overline{x}_n$ to all agents

ADMM illustrated

Set $\beta_n = \lambda_n/\rho$

Algorithm (see e.g. [Boyd'11])

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \overline{x}_n \\ x_{n+1}(v) &= \text{prox}_{\frac{1}{\rho} f_v}\left(\overline{x}_n - \beta_n(v)\right) \end{aligned}$$



4. Compute $\beta_n(v)$, $x_{n+1}(v)$ for all v

Remarks

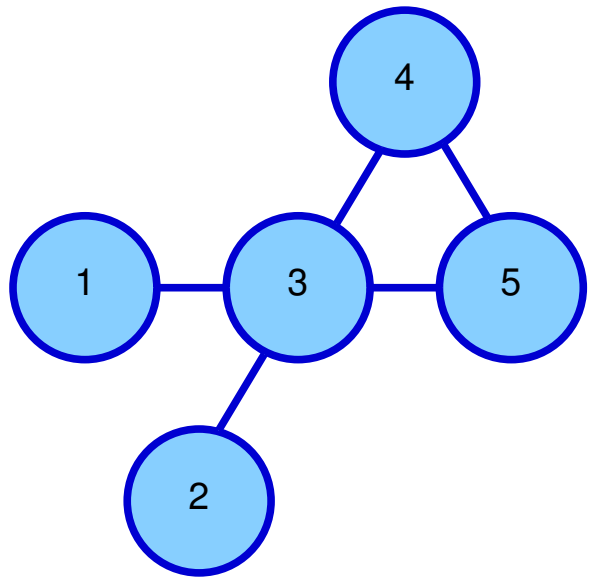
- The algorithm is *parallel* but not *distributed* on the graph
- The algorithm is synchronous

Outline

Context

Consider a non-directed connected graph $\mathcal{G} = (V, \mathcal{E})$

$\mathcal{E} \subset 2^V$ is a set of nondirected edges

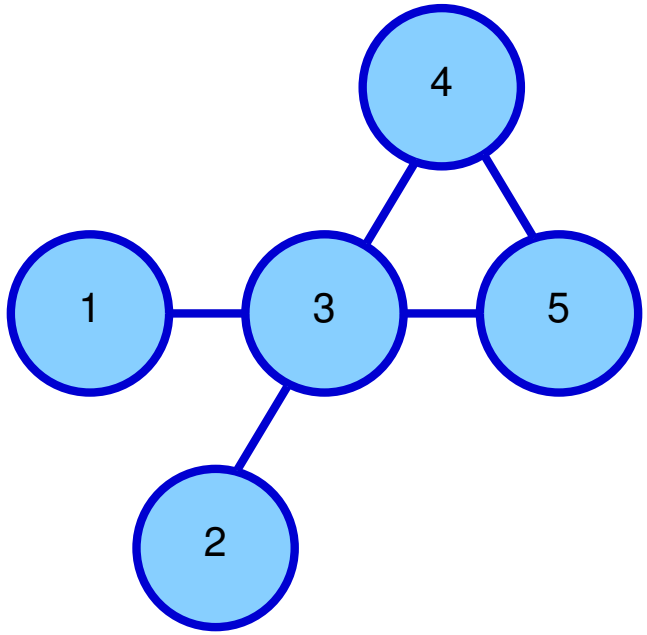


$$\inf_{x \in X^N} F(x) + \iota_{\mathrm{sp}(\mathbf{1})}(x)$$

How to rewrite the penalty $\iota_{\mathrm{sp}(\mathbf{1})}(x)$ to include the graph structure?

Subgraph consensus (Ribeiro et al.'06)

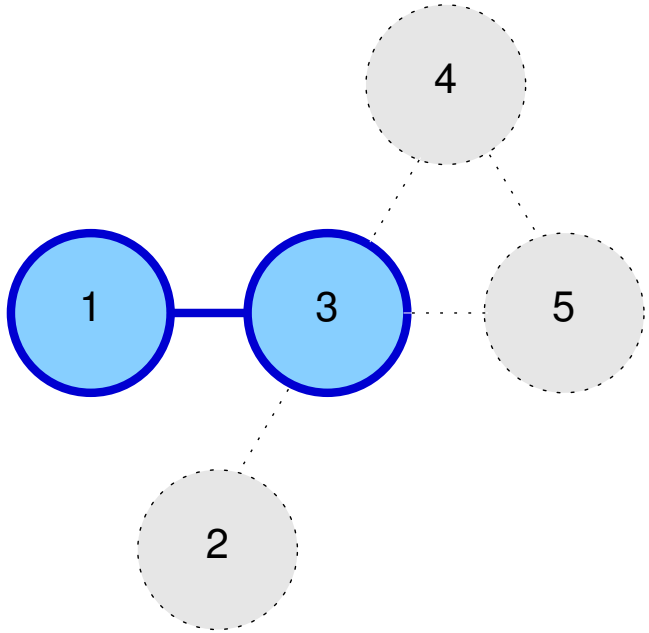
Let $A_1, A_2, \cdots, A_L$ be subsets of V



$A_1 = \{1, 3\}$, $A_2 = \{2, 3\}$, $A_3 = \{3, 4, 5\}$

Subgraph consensus (Ribeiro et al.'06)

Let $A_1, A_2, \cdots, A_L$ be subsets of V

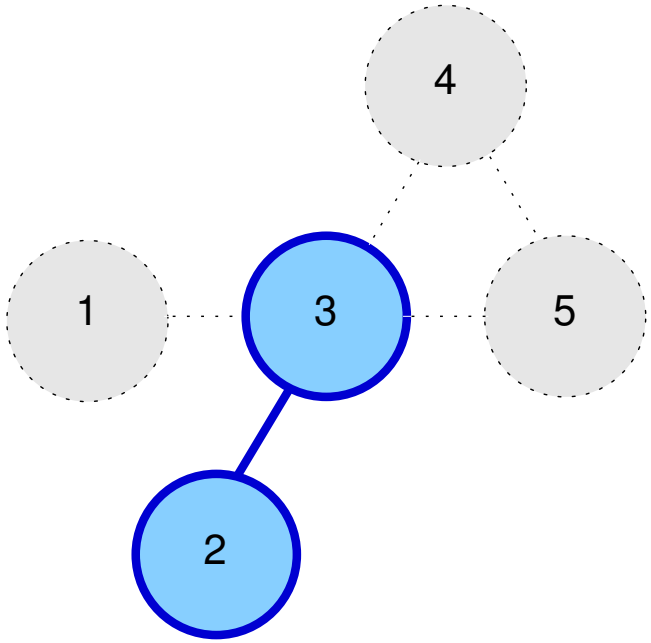


$$\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$A_1 = \{1, 3\}$, $A_2 = \{2, 3\}$, $A_3 = \{3, 4, 5\}$

Subgraph consensus (Ribeiro et al.'06)

Let $A_1, A_2, \cdots, A_L$ be subsets of V



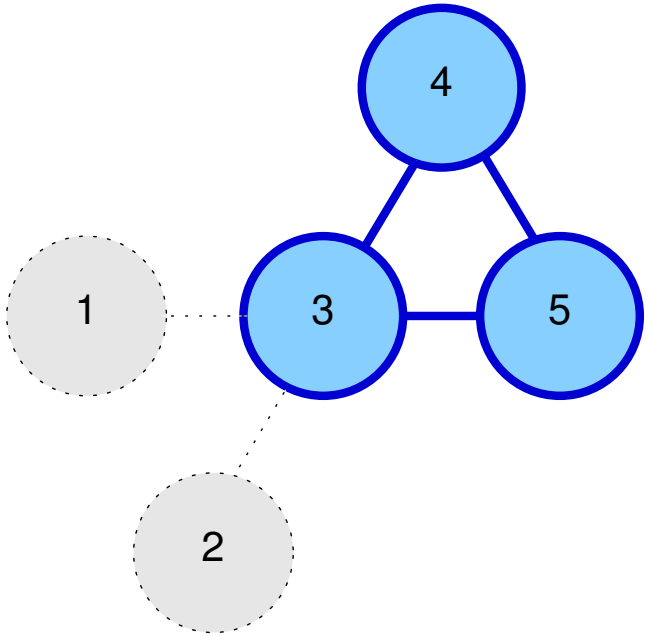
$$\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$$\begin{pmatrix} x(2) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$A_1 = \{1, 3\}$, $A_2 = \{2, 3\}$, $A_3 = \{3, 4, 5\}$

Subgraph consensus (Ribeiro et al.'06)

Let $A_1, A_2, \cdots, A_L$ be subsets of V



$$\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} \in \mathrm{sp}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

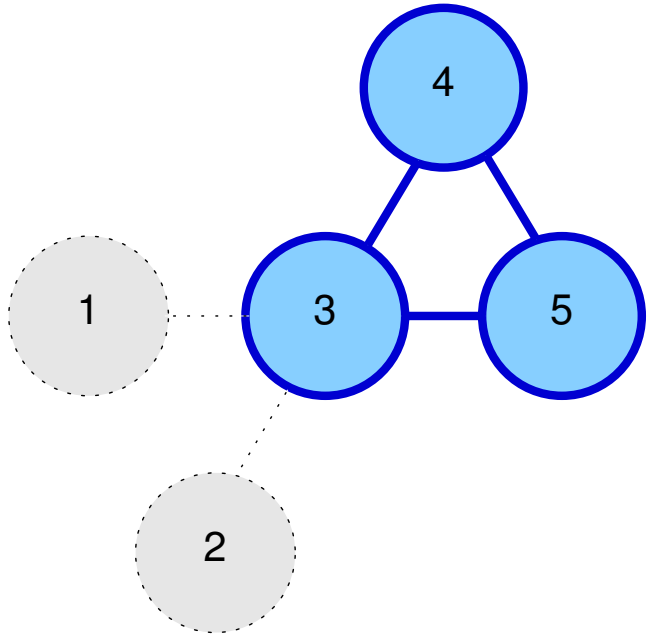
$$\begin{pmatrix} x(2) \\ x(3) \end{pmatrix} \in \mathrm{sp}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x(3) \\ x(4) \\ x(5) \end{pmatrix} \in \mathrm{sp}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$A_1 = \{1, 3\}$, $A_2 = \{2, 3\}$, $A_3 = \{3, 4, 5\}$

Subgraph consensus (Ribeiro et al.'06)

Let $A_1, A_2, \cdots, A_L$ be subsets of V



$$\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$$\begin{pmatrix} x(2) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$$\begin{pmatrix} x(3) \\ x(4) \\ x(5) \end{pmatrix} \in \mathrm{sp}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

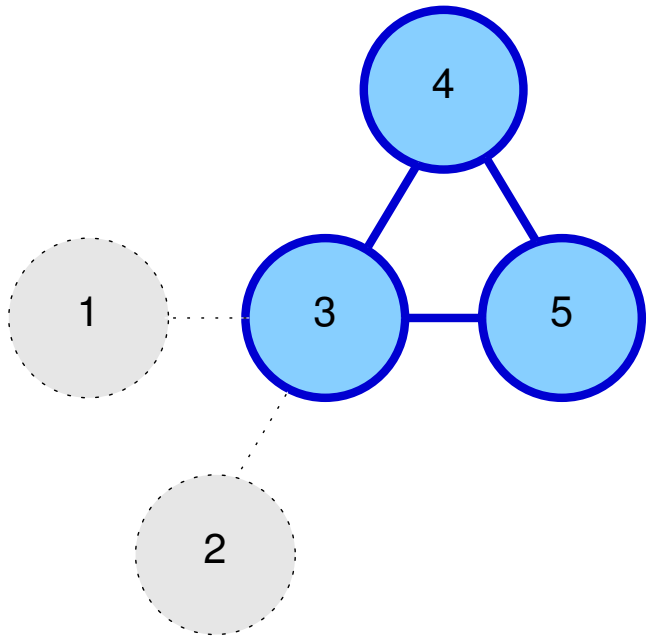
$A_1 = \{1, 3\}$, $A_2 = \{2, 3\}$, $A_3 = \{3, 4, 5\}$

Penalty function

$$\iota_{\mathrm{sp}\binom{1}{1}}\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} + \iota_{\mathrm{sp}\binom{1}{1}}\begin{pmatrix} x(2) \\ x(3) \end{pmatrix} + \iota_{\mathrm{sp}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}}\begin{pmatrix} x(3) \\ x(4) \\ x(5) \end{pmatrix}$$

Subgraph consensus (Ribeiro et al.'06)

Let $A_1, A_2, \cdots, A_L$ be subsets of V



$$\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$$\begin{pmatrix} x(2) \\ x(3) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}$$

$$\begin{pmatrix} x(3) \\ x(4) \\ x(5) \end{pmatrix} \in \mathrm{sp}\binom{1}{1}{1}$$

$A_1 = \{1, 3\}$, $A_2 = \{2, 3\}$, $A_3 = \{3, 4, 5\}$

Penalty function

$$\iota_{\mathrm{sp}\binom{1}{1}}\begin{pmatrix} x(1) \\ x(3) \end{pmatrix} + \iota_{\mathrm{sp}\binom{1}{1}}\begin{pmatrix} x(2) \\ x(3) \end{pmatrix} + \iota_{\mathrm{sp}\binom{1}{1}{1}}\begin{pmatrix} x(3) \\ x(4) \\ x(5) \end{pmatrix} = \iota_{\mathrm{sp}(\mathbf{1})}(x)$$

consensus within subgraphs $\Leftrightarrow$ global consensus

Example (Cont.)

The consensus problem is

$$\inf_{x \in X^N} F(x) + G(Mx)$$

where $Mx = \begin{pmatrix} x(1) \\ x(3) \\ x(2) \\ x(3) \\ x(3) \\ x(4) \\ x(5) \end{pmatrix}$ that is: $M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

and where G is the indicator function of the subspace of vectors of the form

$$\begin{pmatrix} \alpha \\ \alpha \\ \beta \\ \beta \\ \delta \\ \delta \\ \delta \end{pmatrix}$$

General case

Denote by A_1, A_2, $\cdots$ A_L a collection of subsets of V. Define

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_L \end{bmatrix}$$

where M_ℓ is a *selection matrix* of size $|A_\ell| \times N$

Let G denote the indicator function of the vectors z of the form

$$\begin{pmatrix} \alpha_1 \mathbf{1}_{|A_1|} \\ \vdots \\ \alpha_L \mathbf{1}_{|A_L|} \end{pmatrix}$$

Consensus problem:

$$\inf_{x \in X^N} F(x) + G(Mx)$$

ADMM iterations

$$
\begin{aligned}
x_{n+1} &= \arg\min_{x \in X^N} F(x) + \frac{\rho}{2}\|Mx - (z_n - \frac{\lambda_n}{\rho})\|^2 \\
z_{n+1} &= \text{prox}_{\frac{1}{\rho}G}(Mx_{n+1} + \frac{\lambda_n}{\rho}) \\
\lambda_{n+1} &= \lambda_n + \rho(Mx_{n+1} - z_{n+1})
\end{aligned}
$$

ADMM iterations

$$
\begin{aligned}
x_{n+1} &= \arg\min_{x \in X^N} F(x) + \frac{\rho}{2}\|Mx - (z_n - \frac{\lambda_n}{\rho})\|^2 \quad \rightarrow \text{ separable} \\
z_{n+1} &= \text{prox}_{\frac{1}{\rho}G}(Mx_{n+1} + \frac{\lambda_n}{\rho}) \quad \rightarrow \text{ projection} \\
\lambda_{n+1} &= \lambda_n + \rho(Mx_{n+1} - z_{n+1})
\end{aligned}
$$

ADMM

ADMM iterations

$$\begin{aligned} x_{n+1} &= \arg\min_{x \in X^N} F(x) + \frac{\rho}{2}\|Mx - (z_n - \frac{\lambda_n}{\rho})\|^2 \\ z_{n+1} &= \text{prox}_{\frac{1}{\rho}G}(Mx_{n+1} + \frac{\lambda_n}{\rho}) \\ \lambda_{n+1} &= \lambda_n + \rho(Mx_{n+1} - z_{n+1}) \end{aligned}$$

Notations

- For all ℓ, $\overline{x}_n^{(\ell)} = \frac{1}{|A_\ell|}\sum_{v \in A_\ell} x_n(v)$ is the ℓth subgraph-average
- For all v, $\sigma_v \subset \{1, \cdots, L\}$ is the set of indices ℓ such that $v \in A_\ell$
- $\chi_n(v) = \frac{1}{|\sigma_v|}\sum_{\ell \in \sigma_v} \overline{x}_n^{(\ell)}$ is the average of subgraphs-averages in σ_v

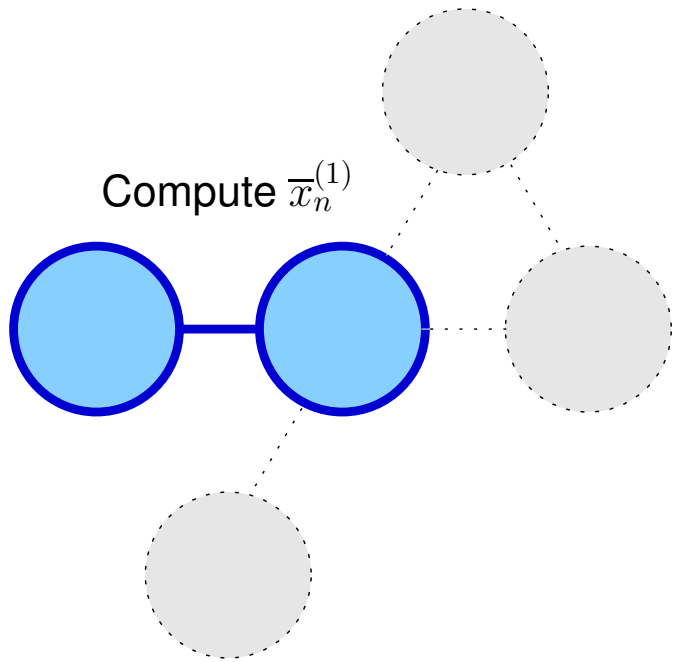
Distributed ADMM (Ribeiro et al.'06)

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \chi_n(v) \\ x_{n+1}(v) &= \text{prox}_{\frac{f_v}{\rho|\sigma_v|}}(\chi_n(v) - \beta_n(v)) \end{aligned}$$

Distributed ADMM illustrated

Distributed ADMM (Ribeiro et al.'06)

$$\text{For all } v, \quad \begin{aligned} \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \chi_n(v) \\ x_{n+1}(v) &= \text{prox}_{\frac{f_v}{\rho|\sigma_v|}}\left(\chi_n(v) - \beta_n(v)\right) \end{aligned}$$

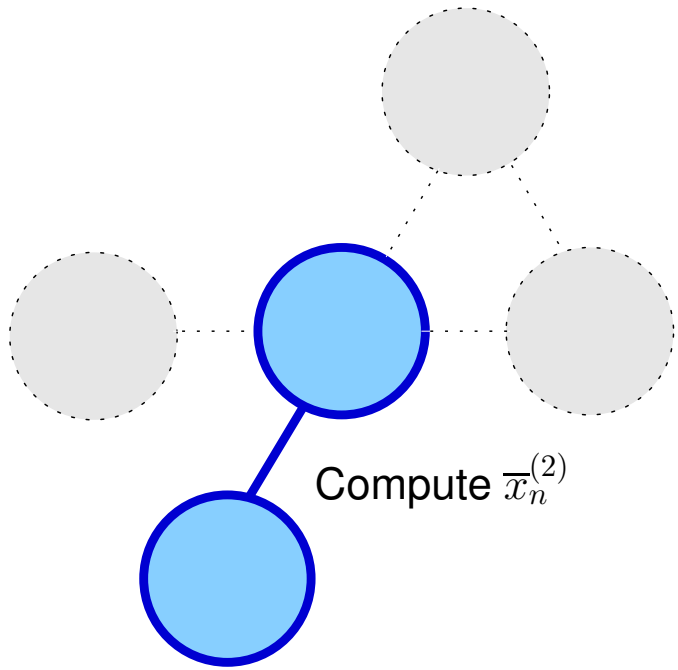


1. For each subgraph, compute average $\overline{x}_n^{(\ell)}$

Distributed ADMM illustrated

Distributed ADMM (Ribeiro et al.'06)

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \chi_n(v) \\ x_{n+1}(v) &= \text{prox}_{\frac{f_v}{\rho|\sigma_v|}}\left(\chi_n(v) - \beta_n(v)\right) \end{aligned}$$

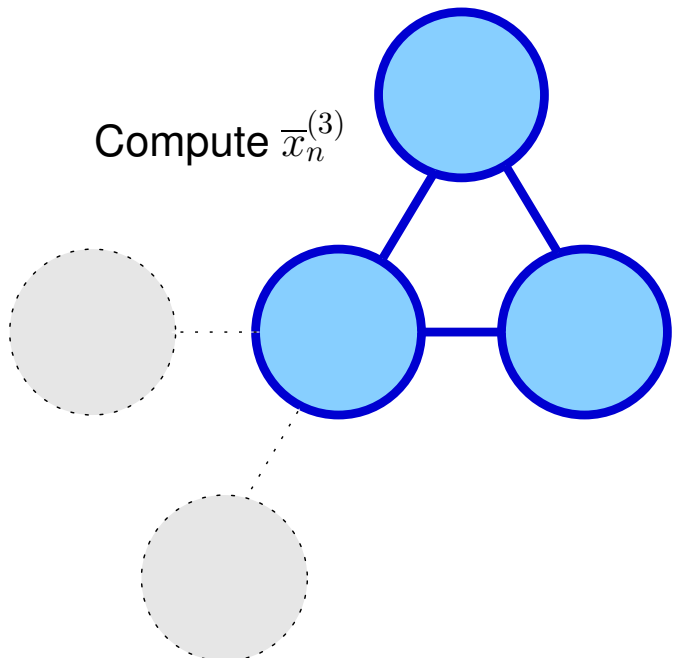


1. For each subgraph, compute average $\overline{x}_n^{(\ell)}$

Distributed ADMM illustrated

Distributed ADMM (Ribeiro et al.'06)

$$\text{For all } v, \quad \begin{aligned} \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \chi_n(v) \\ x_{n+1}(v) &= \text{prox}_{\frac{f_v}{\rho|\sigma_v|}}\left(\chi_n(v) - \beta_n(v)\right) \end{aligned}$$

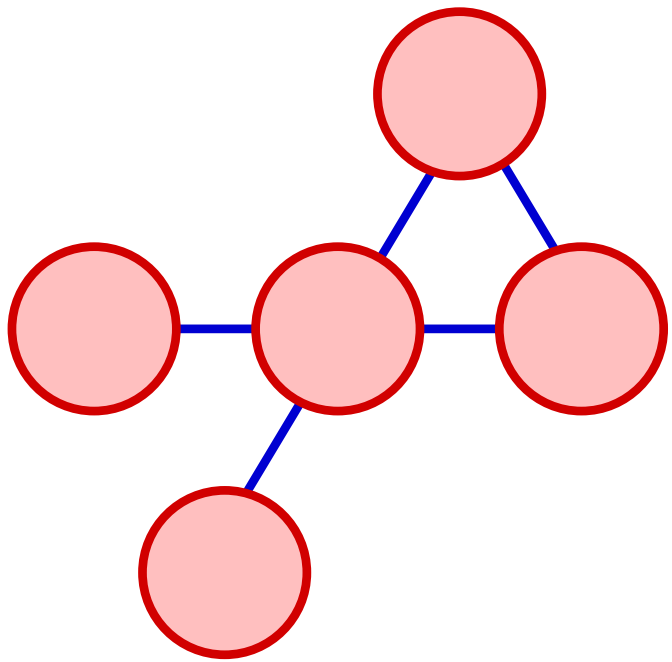


1. For each subgraph, compute average $\overline{x}_n^{(\ell)}$

Distributed ADMM illustrated

Distributed ADMM (Ribeiro et al.'06)

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \chi_n(v) \\ x_{n+1}(v) &= \text{prox}_{\frac{f_v}{\rho|\sigma_v|}}\left(\chi_n(v) - \beta_n(v)\right) \end{aligned}$$

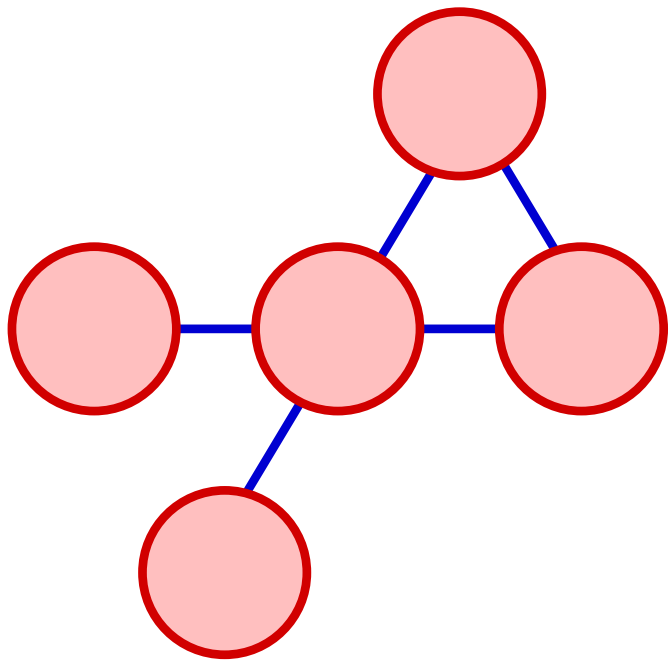


2. For each vertex v, compute $\chi_n(v) = \text{Average}(\overline{x}_n^{(\ell)} : v \in A_\ell)$

Distributed ADMM illustrated

Distributed ADMM (Ribeiro et al.'06)

$$\begin{aligned} \text{For all } v, \quad \beta_n(v) &= \beta_{n-1}(v) + x_n(v) - \chi_n(v) \\ x_{n+1}(v) &= \text{prox}_{\frac{f_v}{\rho|\sigma_v|}}\left(\chi_n(v) - \beta_n(v)\right) \end{aligned}$$



3. For each vertex v, compute $\beta_n(v)$ and $x_{n+1}(v)$

The burden of synchronism

- All agents must complete their prox before combining
- The network waits for the slowest agents

Our objective for now on: allow for asynchronism

Outline

Monotone operators

A **monotone operator** is a set-valued application $A : X \to 2^X$ such that

$$\forall (x, y),\ \forall (u, v) \in A(x) \times A(y), \quad \langle u - v, x - y \rangle \geq 0$$

- It is **maximal** if it is not contained in an other monotone operator
- A point x is a **zero** of A if $0 \in A(x)$
- We identify A with its graph $\{(x, u) : x \in X, u \in A(x)\}$

Monotone operators

A **monotone operator** is a set-valued application $A : X \to 2^X$ such that

$$\forall (x, y),\ \forall (u, v) \in A(x) \times A(y), \quad \langle u - v, x - y \rangle \geq 0$$

- It is **maximal** if it is not contained in an other monotone operator
- A point x is a **zero** of A if $0 \in A(x)$
- We identify A with its graph $\{(x, u) : x \in X, u \in A(x)\}$

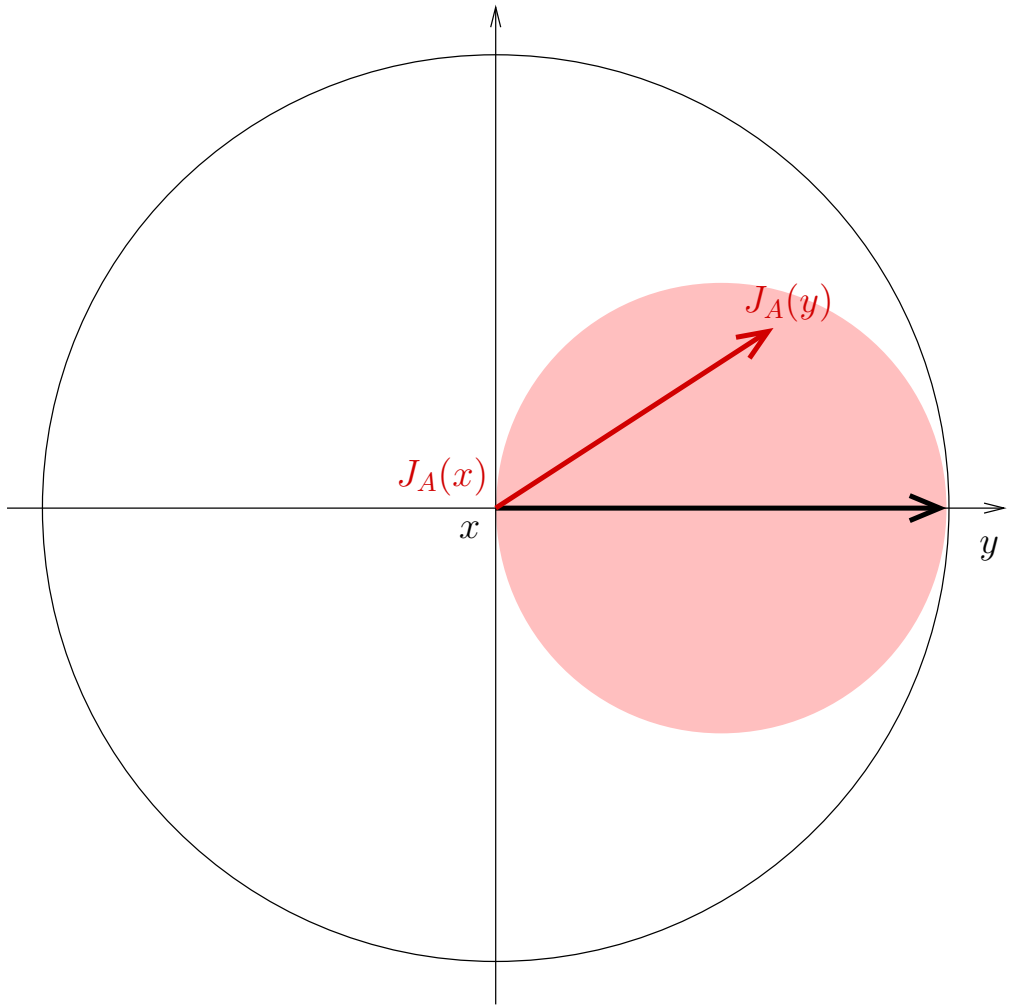
The **resolvent** of A is

$$J_A = (I + A)^{-1}$$

- $\mathrm{dom}(J_A) = X$ whenever A is maximal
- J_A is single-valued (it is a function)
- a fixed point of J_A is a zero of A

Firm non expansiveness

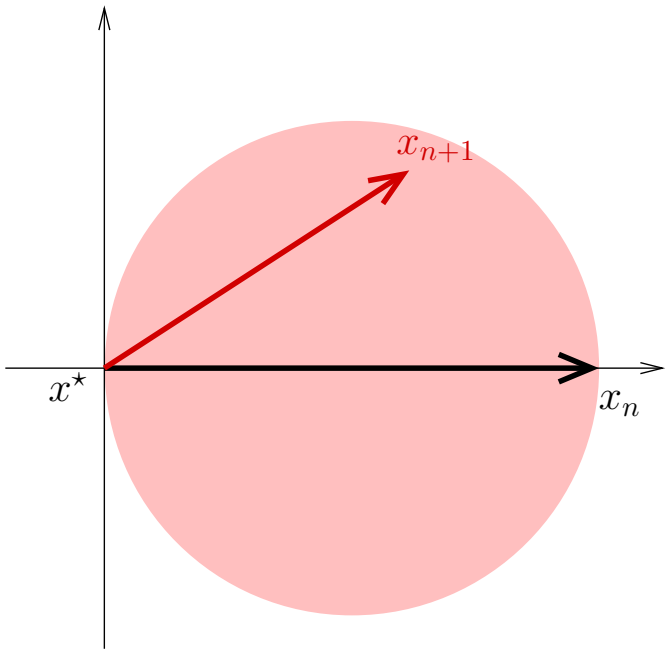
$$\forall x, y \in \mathrm{dom}(J_A), \quad \langle J_A(x) - J_A(y), x - y \rangle \geq \|J_A(x) - J_A(y)\|^2$$



proximal point algorithm

$$x_{n+1} = J_A(x_n)$$

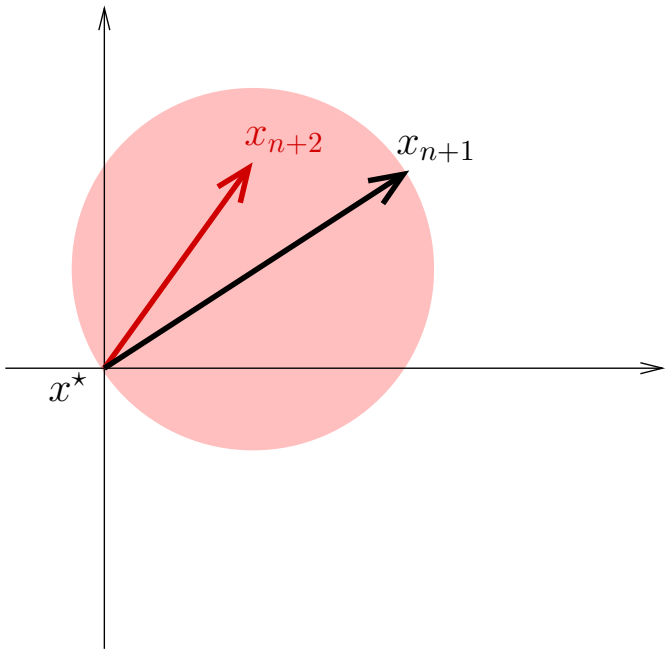
Assume that there exists $x^\star \in \mathrm{Zer}(A)$



proximal point algorithm

$$x_{n+1} = J_A(x_n)$$

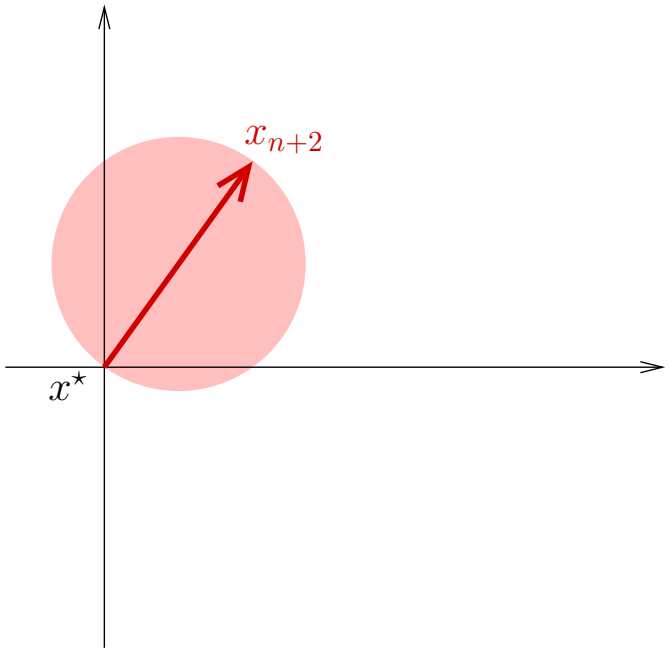
Assume that there exists $x^\star \in \mathrm{Zer}(A)$



proximal point algorithm

$$x_{n+1} = J_A(x_n)$$

Assume that there exists $x^\star \in \mathrm{Zer}(A)$



$\|x_n - x^\star\|$ decreases with n

Convergence of the proximal point algorithm

If A is maximal monotone and $\mathrm{Zer}(A) \neq \emptyset$, x_n converges to a point in $\mathrm{Zer}(A)$

Douglas-Rachford (DR) operator

Problem: Find a zero of the sum of two monotone operators $A + B$

Douglas Rachford operator:

$$S = \{(v + \rho b, u - v) \ : \ (u, b) \in B, \ (v, a) \in A, \ u + \rho a = v - \rho b\}$$

Property: If $\zeta^\star \in \mathrm{Zer}(S)$, then $J_{\rho B}(\zeta^\star) \in \mathrm{Zer}(A + B)$

Douglas-Rachford algorithm

Let A, B maximal monotone such that $\mathrm{Zer}(A + B) \neq \emptyset$. Set

$$\zeta_{n+1} \quad = \quad J_S(\zeta_n)$$

Then $\lambda_n = J_{\rho B}(\zeta_n)$ converges to a point in $\mathrm{Zer}(A + B)$

ADMM as a Douglas-Rachford algorithm

Consider the problem

$$\inf_{x \in X^N} F(x) + G(Mx)$$

Under mild qualification conditions, the infimum coincides with

$$\min_{\lambda} F^\star(-M^\star \lambda) + G^\star(\lambda)$$

Solving the above optimization problem $=$ finding a zero of

$$\partial \left[F^\star \circ (-M^\star) + G^\star \right]$$

ADMM as a Douglas-Rachford algorithm

Consider the problem

$$\inf_{x \in X^N} F(x) + G(Mx)$$

Under mild qualification conditions, the infimum coincides with

$$\min_{\lambda} F^\star(-M^\star \lambda) + G^\star(\lambda)$$

Solving the above optimization problem $=$ finding a zero of

$$-M\partial F^\star \circ (-M^\star) + \partial G^\star$$

ADMM as a Douglas-Rachford algorithm

Consider the problem

$$\inf_{x \in X^N} F(x) + G(Mx)$$

Under mild qualification conditions, the infimum coincides with

$$\min_{\lambda} F^\star(-M^\star \lambda) + G^\star(\lambda)$$

Solving the above optimization problem = finding a zero of

$$-M\partial F^\star \circ (-M^\star) + \partial G^\star$$

The Douglas-Rachford algorithm boils down to ADMM when

$$\begin{aligned} A &= -M\partial F^\star \circ (-M^\star) \\ B &= \partial G^\star \end{aligned}$$

Block-components

Notation: denote by $\zeta^{(\ell)}$ the ℓth block-component of $\zeta \in X^{|A_1|+\cdots+|A_L|}$

$$\zeta = \begin{pmatrix} \zeta^{(1)} \\ \vdots \\ \zeta^{(L)} \end{pmatrix} \quad \text{where } \zeta^{(\ell)} = (\zeta^{(\ell)}(v))_{v \in A_\ell}$$

Let S be the DR operator of two maximal monotone operators A and B

Douglas-Rachford algorithm

$$\zeta_{n+1} = \begin{pmatrix} J_S^{(1)}(\zeta_n) \\ \vdots \\ J_S^{(L)}(\zeta_n) \end{pmatrix}$$

This means that $\zeta_{n+1}^{(\ell)} = J_S^{(\ell)}(\zeta_n)$ for all $\ell = 1, \ldots, L$

Asynchronous Douglas-Rachford algorithm

Asynchronous Douglas-Rachford algorithm

At time n, select a subgraph $\ell \in \{1, \ldots, L\}$ at random. Set

$$\begin{aligned} \zeta_{n+1}^{(\ell)} &= J_S^{(\ell)}(\zeta_n) \\ \zeta_{n+1}^{(k)} &= \zeta_n^{(k)} \text{ for all } k \neq \ell \end{aligned}$$

We should

- prove that this 'degraded' algorithm still converges to $\mathrm{Zer}(S)$
- make the implementation explicit

Convergence

Denote by ζ_n the sequence generated by the asynchronous DR algorithm

Assumptions

- The indices of the active subgraph at time n forms an iid sequence
- $\mathrm{Zer}(A+B) \neq \emptyset$

Theorem (Iutzeler et al.'13, submitted)

Sequence ζ_n converges almost surely to a random variable supported by $\mathrm{Zer}(S)$

Corollary

Sequence $\lambda_n = J_{\rho B}(\zeta_n)$ converges a.s. to a r.v. supported by $\mathrm{Zer}(A+B)$

Asynchronous algorithm explicited (1/3)

The consensus problem can be formulated as

$$\inf_{x \in X^N} F(x) + G(Mx)$$

Example:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad G = \text{indicator of vectors of the form } \begin{pmatrix} \alpha \\ \alpha \\ \beta \\ \beta \\ \delta \\ \delta \\ \delta \end{pmatrix}$$

Let us explicit the asynchronous DR algorithm for the monotone operators

$$A = -M \partial F^\star \circ (-M^\star) \quad \text{and} \quad B = \partial G^\star$$

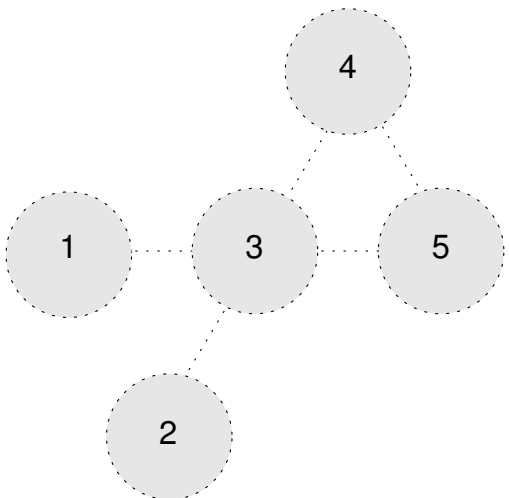
Asynchronous ADMM explicited (2/3)

General case: Denote by A_1, A_2, $\cdots$ A_L a collection of subgraphs

Each node v maintains the variables

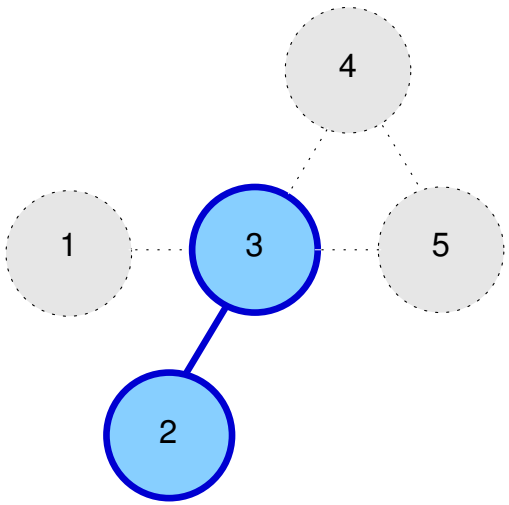
$$x_n(v),\ \lambda_n^{(\ell)}(v),\ \overline{z}_n^{(\ell)} \quad \forall \ell \text{ such that } v \in A_\ell$$

Asynchronous ADMM explicited (3/3)

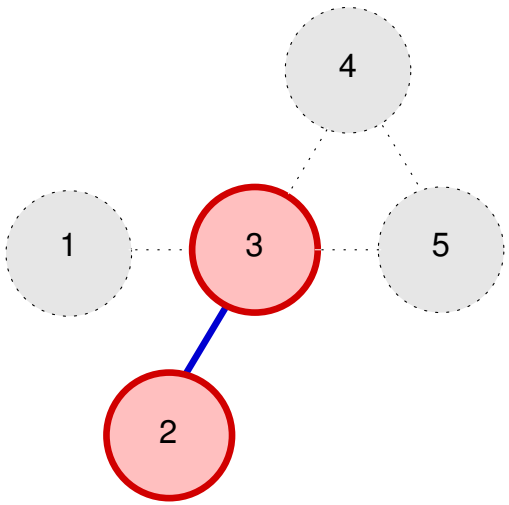


Asynchronous ADMM explicited (3/3)

At time n, a component A_ℓ is activated



Asynchronous ADMM explicited (3/3)

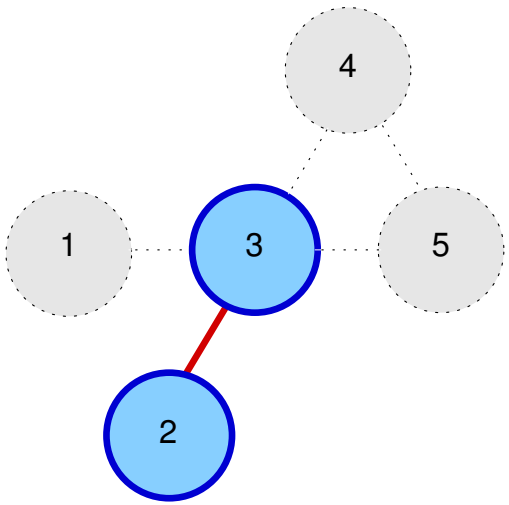


At time n, a component A_ℓ is activated

All agents $v \in A_\ell$ compute

$$x_{n+1}(v) = \text{prox}_{\frac{f_v}{\rho|\sigma_v|}} \left(\frac{1}{|\sigma_v|} \sum_{k \in \sigma_v} \left(\bar{z}_n^{(k)} - \frac{\lambda_n^{(k)}(v)}{\rho} \right) \right)$$

Asynchronous ADMM explicited (3/3)



At time n, a component A_ℓ is activated

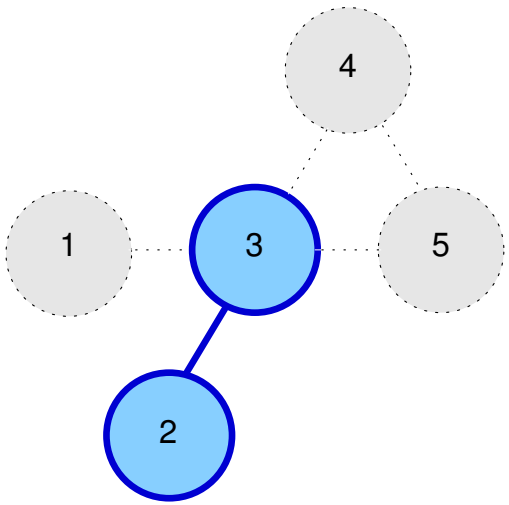
All agents $v \in A_\ell$ compute

$$x_{n+1}(v) = \text{prox}_{\frac{f_v}{\rho|\sigma_v|}} \left(\frac{1}{|\sigma_v|} \sum_{k \in \sigma_v} \left(\bar{z}_n^{(k)} - \frac{\lambda_n^{(k)}(v)}{\rho} \right) \right)$$

All agents in A_ℓ communicate to find the average

$$\bar{z}_{n+1}^{(\ell)} = \frac{1}{|A_\ell|} \sum_{w \in A_\ell} x_{n+1}(w)$$

Asynchronous ADMM explicited (3/3)



At time n, a component A_ℓ is activated

All agents $v \in A_\ell$ compute

$$x_{n+1}(v) = \text{prox}_{\frac{f_v}{\rho|\sigma_v|}} \left(\frac{1}{|\sigma_v|} \sum_{k \in \sigma_v} \left(\bar{z}_n^{(k)} - \frac{\lambda_n^{(k)}(v)}{\rho} \right) \right)$$

All agents in A_ℓ communicate to find the average

$$\bar{z}_{n+1}^{(\ell)} = \frac{1}{|A_\ell|} \sum_{w \in A_\ell} x_{n+1}(w)$$

All agents $v \in A_\ell$ update

$$\lambda_{n+1}^{(\ell)}(v) = \lambda_n^{(\ell)}(v) + \rho(x_{n+1}(v) - \bar{z}_{n+1}^{(\ell)})$$

Other variables are maintained to former values