Learning to control networked-coupled subsystems with unknown dynamics

Aditya Mahajan

McGill University

Mean Field Games on Networks Workshop
28th Oct 2021

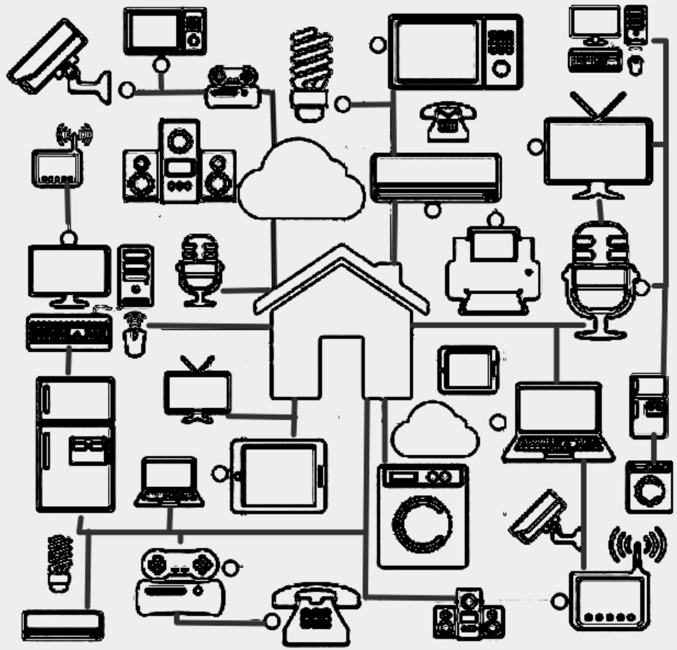
- email: aditya.mahajan@mcgill.ca
- homepage: http://cim.mcgill.ca/~adityam

Networks are ubiquitous

Networks are ubiquitous

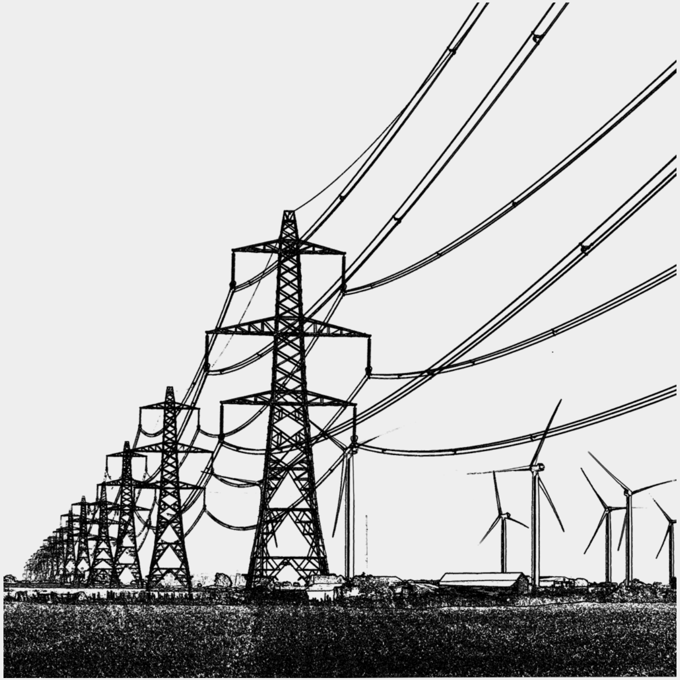
Transportation networks

Networks are ubiquitous



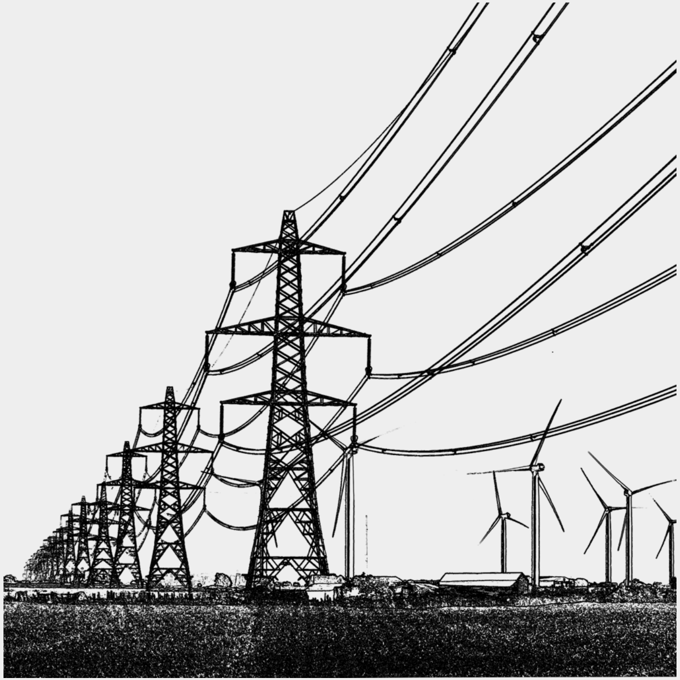
Internet of Things

Networks are ubiquitous



Energy network

Networks are ubiquitous

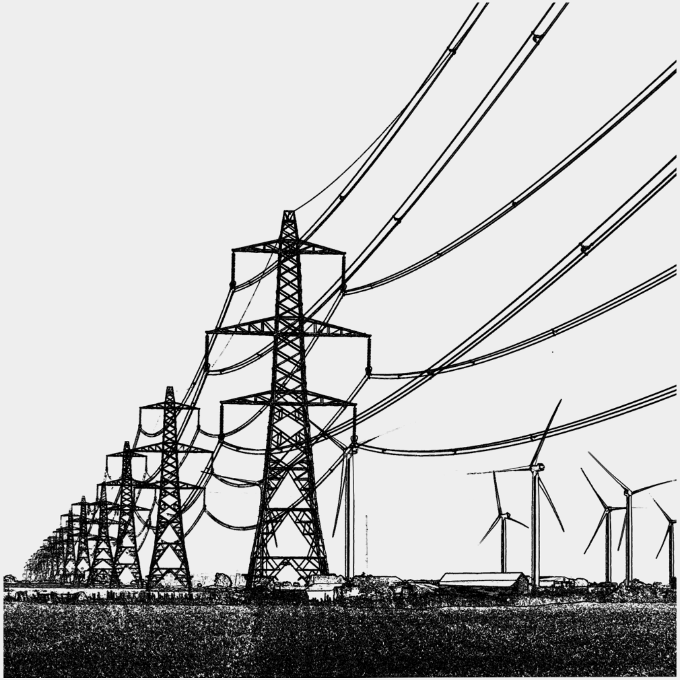


Energy network

Salient Features

- Large/growing size
- Nodes have local states
- Coupled dynamics and costs

Networks are ubiquitous



Energy network

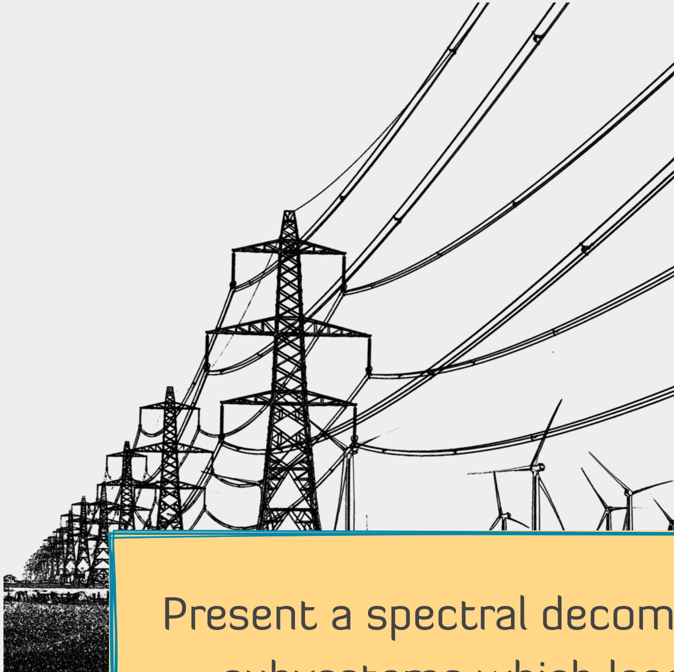
Salient Features

- Large/growing size
- Nodes have local states
- Coupled dynamics and costs

Design challenges

- Scalability of the solution
- How to handle model uncertainty

Networks are ubiquitous



Salient Features

- Large/growing size
- Nodes have local states
- Coupled dynamics and costs

Design challenges

- Scalability of the solution
- How to handle model uncertainty

Present a spectral decomposition method for network-coupled subysstems which leads to scalable planning and learning

Outline

System Model

- Network-coupled subsystems
  - Agents interacting over a graph
  - Coupled dynamics
  - Coupled costs

Outline

System Model

- Network-coupled subsystems
  - Agents interacting over a graph
  - Coupled dynamics
  - Coupled costs

Planning solution

- Spectral factorization of dynamics and cost
- Decoupled Riccati equations

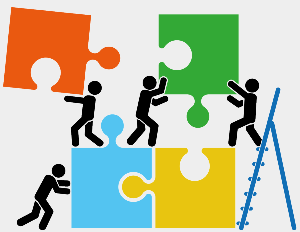
Outline

System Model

- Network-coupled subsystems
  - Agents interacting over a graph
  - Coupled dynamics
  - Coupled costs

Planning solution

- Spectral factorization of dynamics and cost
- Decoupled Riccati equations



Learning solution

- Spectral factorization of learning
- Numerical experiments

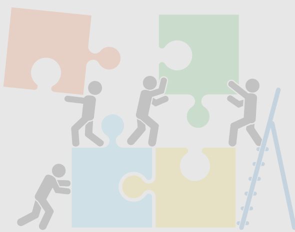
Outline

System Model

- Network-coupled subsystems
  - Agents interacting over a graph
  - Coupled dynamics
  - Coupled costs

Planning solution

- Spectral factorization of dynamics and cost
- Decoupled Riccati equations



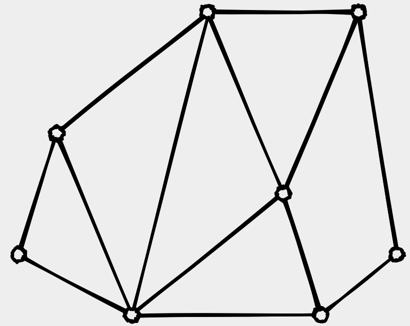
Learning solution

- Spectral factorization of learning
- Numerical experiments

System Model

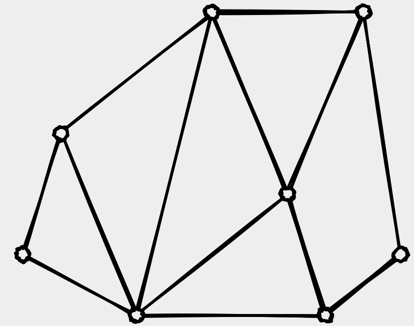
Weighted undirected graph $\mathcal{G}$

- Nodes $N = \{1, \dots, n\}$.
- Symmetric matrix $M = [m^{ij}]$ associated with $\mathcal{G}$
  (e.g., weighted adjacency matrix, weighted Laplacian, etc.)



System Model

Weighted undirected graph $\mathcal{G}$



- Nodes $N = \{1, \dots, n\}$.
- Symmetric matrix $M = [m^{ij}]$ associated with $\mathcal{G}$ (e.g., weighted adjacency matrix, weighted Laplacian, etc.)

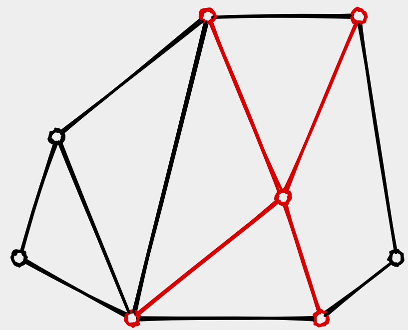
System Dynamics

- A subsystem located at each node. State $x^i_t \in \mathbb{R}^{d_x}$. Control $u^i_t \in \mathbb{R}^{d_u}$.

$$x^i_{t+1} = Ax^i_t + Bu^i_t + D\sum_{j \in N} m^{ij} x^j_t + E \sum_{j \in N} m^{ij} u^j_t + w^i_t$$

System Model

Weighted undirected graph $\mathcal{G}$



- Nodes $N = \{1, \dots, n\}$.
- Symmetric matrix $M = [m^{ij}]$ associated with $\mathcal{G}$
  (e.g., weighted adjacency matrix, weighted Laplacian, etc.)

System Dynamics

- A subsystem located at each node. State $x^i_t \in \mathbb{R}^{d_x}$. Control $u^i_t \in \mathbb{R}^{d_u}$.

$$x^i_{t+1} = Ax^i_t + Bu^i_t + D\sum_{j\in N} m^{ij}x^j_t + E\sum_{j\in N} m^{ij}u^j_t + w^i_t$$

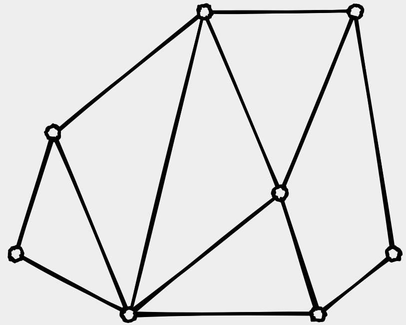
Network field of states $x^{\mathcal{G},i}_t$ **Network field** of control $u^{\mathcal{G},i}_t$

System Model (cont.)

Per-step cost

$$c(x_t, u_t) = \sum_{i,j \in N} \left[h_q^{ij} (x_t^i)^\top Q (x_t^j) + h_r^{ij} (u_t^i)^\top Q (u_t^j) \right]$$

where $H_q = [h_q^{ij}]$ and $H_r = [h_r^{ij}]$ are symmetric matrices which have the same eigenvectors as M.

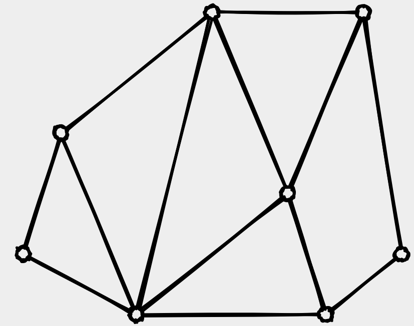


System Model (cont.)

Per-step cost

$$c(x_t, u_t) = \sum_{i,j \in N} \left[h_q^{ij} (x_t^i)^\top Q (x_t^j) + h_r^{ij} (u_t^i)^\top Q (u_t^j) \right]$$

where $H_q = [h_q^{ij}]$ and $H_r = [h_r^{ij}]$ are symmetric matrices which have the same eigenvectors as M.

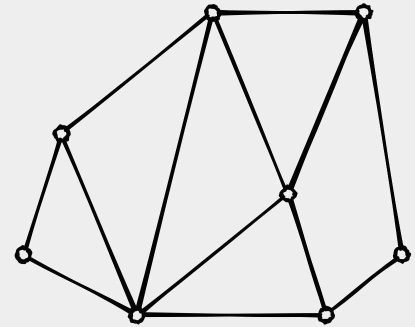


Remark

For two symmetric $n \times n$ matrices M_1 and M_2, the following statements are equivalent:

- M_1 and M_2 share the same eigenvectors.
- M_1 and M_2 communte (i.e., $M_1 M_2 = M_2 M_1$)
- M_1 and M_2 are simultaneously diagonalizable.

System Model (cont.)



Per-step cost

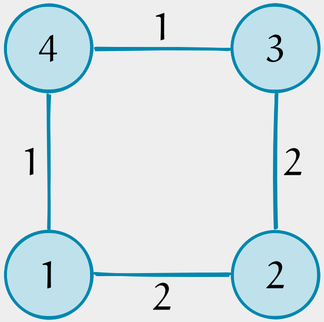
$$c(x_t, u_t) = \sum_{i,j \in N} \left[h_q^{ij} (x_t^i)^\top Q (x_t^j) + h_r^{ij} (u_t^i)^\top Q (u_t^j) \right]$$

where $H_q = [h_q^{ij}]$ and $H_r = [h_r^{ij}]$ are symmetric matrices which have the same eigenvectors as M.

Important special case

- $H_q = \sum_{k=0}^{K_q} q_k M^k$ and $H_r = \sum_{k=0}^{K_r} r_k M^k$.
- Captures the intuition that the per-step cost respects the graph structure.
- Example: $H_q = q_0 I + q_1 M + q_2 M^2$ means that there is a cost coupling between the one- and two-hop neighbors.

An example to illustrate that nodes are not exchageable



A graph $\mathcal{G}$

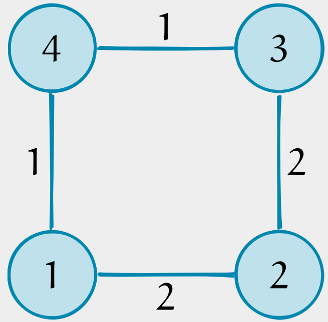
An example to illustrate that nodes are not exchageable

Dynamical coupling

- Nodes are not exchageable

$$x_t^{\mathcal{G},1} = 2\, x_t^2 + 1\, x_t^4, \qquad x_t^{\mathcal{G},2} = 2\, x_t^1 + 2\, x_t^3,$$

$$x_t^{\mathcal{G},3} = 2\, x_t^2 + 1\, x_t^4, \qquad x_t^{\mathcal{G},4} = 1\, x_t^1 + 1\, x_t^3.$$



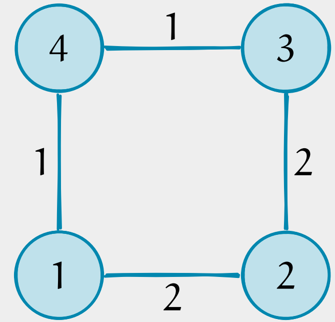
A graph $\mathcal{G}$

An example to illustrate that nodes are not exchageable

Dynamical coupling

- Nodes are not exchageable

$$x_t^{\mathcal{G},1} = 2\, x_t^2 + 1\, x_t^4, \qquad x_t^{\mathcal{G},2} = 2\, x_t^1 + 2\, x_t^3,$$
$$x_t^{\mathcal{G},3} = 2\, x_t^2 + 1\, x_t^4, \qquad x_t^{\mathcal{G},4} = 1\, x_t^1 + 1\, x_t^3.$$



A graph $\mathcal{G}$

Cost coupling

- Nodes are not exchageable

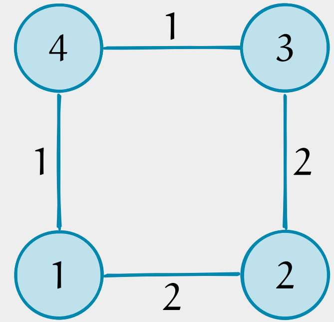
Suppose $H_q = q_0 I + q_1 M + q_2 M^2$.

An example to illustrate that nodes are not exchageable

Dynamical coupling

- Nodes are not exchageable

$$x_t^{\mathcal{G},1} = 2\,x_t^2 + 1\,x_t^4, \qquad x_t^{\mathcal{G},2} = 2\,x_t^1 + 2\,x_t^3,$$
$$x_t^{\mathcal{G},3} = 2\,x_t^2 + 1\,x_t^4, \qquad x_t^{\mathcal{G},4} = 1\,x_t^1 + 1\,x_t^3.$$

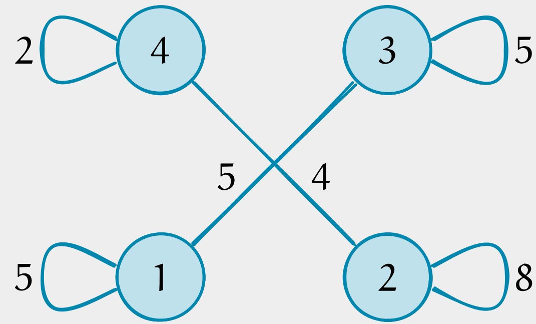


A graph $\mathcal{G}$

Cost coupling

- Nodes are not exchageable

Suppose $H_q = q_0 I + q_1 M + q_2 M^2$.



Two-hop neighborhood

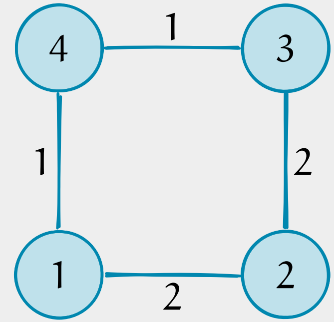
An example to illustrate that nodes are not exchageable

Dynamical coupling

- Nodes are not exchageable

$$x_t^{\mathcal{G},1} = 2\,x_t^2 + 1\,x_t^4, \qquad x_t^{\mathcal{G},2} = 2\,x_t^1 + 2\,x_t^3,$$

$$x_t^{\mathcal{G},3} = 2\,x_t^2 + 1\,x_t^4, \qquad x_t^{\mathcal{G},4} = 1\,x_t^1 + 1\,x_t^3.$$



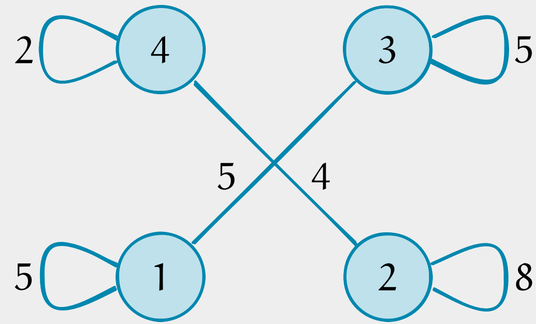
A graph $\mathcal{G}$

Cost coupling

- Nodes are not exchageable

Suppose $H_q = q_0 I + q_1 M + q_2 M^2$. Then

$$H_q = \begin{bmatrix} q_0 + 5q_2 & 2q_1 & 5q_2 & q_1 \\ 2q_1 & q_0 + 8q_2 & 2q_1 & 4q_2 \\ 5q_2 & 2q_1 & q_0 + 5q_2 & q_1 \\ q_1 & 4q_2 & q_1 & q_0 + 2q_2 \end{bmatrix}$$

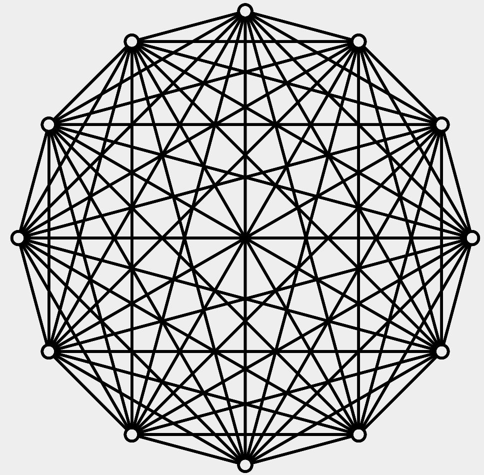


Two-hop neighborhood

Model generalizes mean-field control model

Special case

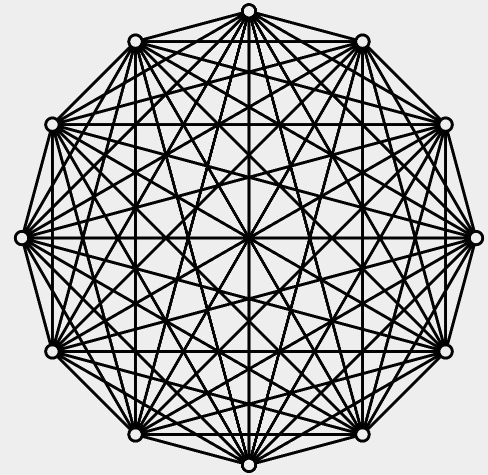
- Consider $M = \frac{1}{n}\mathbb{1}_{n\times n}$ and $H_q = H_r = \frac{1}{n}I + \frac{\kappa}{n}M$.
- Network-field $\frac{1}{n}\sum_{j\in N} x_t^j =: \bar{x}_t$ is the (empirical) mean-field.



Model generalizes mean-field control model

Special case

- Consider $M = \frac{1}{n}\mathbb{1}_{n\times n}$ and $H_q = H_r = \frac{1}{n}I + \frac{\kappa}{n}M$.
- Network-field $\frac{1}{n}\sum_{j\in N} x^j_t =: \bar{x}_t$ is the (empirical) mean-field.



Dynamics

$$x^i_{t+1} = Ax^i_t + Bu^i_t + D\bar{x}_t + E\bar{u}_t + w^i_t.$$

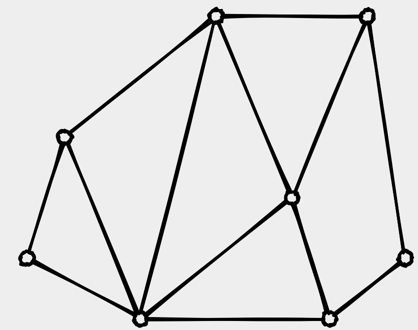
Per-step cost

$$c(x_t, u_t) = (1+\kappa)\big[\bar{x}_t^\top Q\bar{x}_t + \bar{u}_t^\top R\bar{u}_t\big] + \frac{1}{n}\sum_{i\in N}\big[(x^i_t - \bar{x}_t)^\top Q(x^i_t - \bar{x}_t) + (u^i_t - \bar{u}_t)^\top Q(u^i_t - \bar{u}_t)\big].$$

Problem formulation

Summary of the model

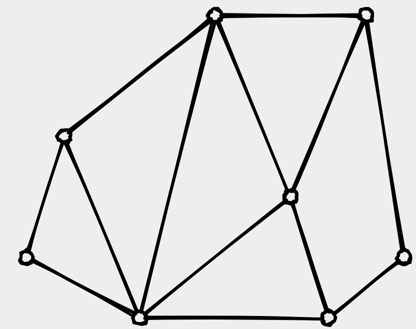
- Dynamics: $x^i_{t+1} = A x^i_t + B u^i_t + D \sum_{j \in N} m^{ij} x^j_t + E \sum_{j \in N} m^{ij} u^j_t + w^i_t$
- Per-step cost: $c(x_t, u_t) = \sum_{i,j \in N} \left[h^{ij}_q (x^i_t)^\top Q (x^j_t) + h^{ij}_r (u^i_t)^\top Q (u^j_t) \right]$
- Network structure: M, H_q, and H_r have the same eigenvectors.



Problem formulation

Summary of the model

- Dynamics: $x^i_{t+1} = Ax^i_t + Bu^i_t + D\sum_{j\in N} m^{ij} x^j_t + E\sum_{j\in N} m^{ij} u^j_t + w^i_t$
- Per-step cost: $c(x_t, u_t) = \sum_{i,j\in N}\left[h^{ij}_q (x^i_t)^\top Q(x^j_t) + h^{ij}_r (u^i_t)^\top Q(u^j_t)\right]$
- Network structure: M, H_q, and H_r have the same eigenvectors.



Objective

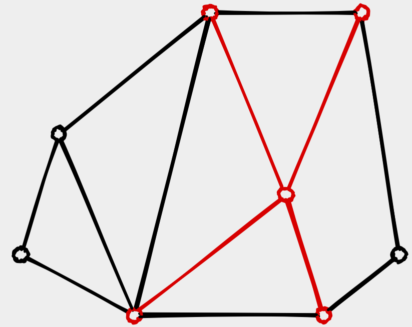
Choose a policy $\pi: (x^1_t, \dots, x^n) \to (u^1_t, \dots, u^n_t)$ to minimize:

$$\limsup_{T\to\infty} \frac{1}{T}\mathbb{E}^\pi\left[\sum_{t=1}^{T} c(x_t, u_t)\right]$$

Problem formulation

Summary of the model

- Dynamics: $x^i_{t+1} = A x^i_t + B u^i_t + D \sum_{j \in N} m^{ij} x^j_t + E \sum_{j \in N} m^{ij} u^j_t + w^i_t$
- Per-step cost: $c(x_t, u_t) = \sum_{i,j \in N} \left[h^{ij}_q (x^i_t)^\top Q (x^j_t) + h^{ij}_r (u^i_t)^\top Q (u^j_t) \right]$
- Network structure: M, H_q, and H_r have the same eigenvectors.



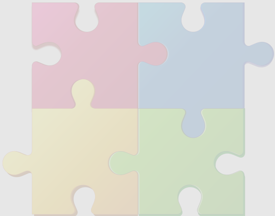
Objective

Choose a policy π: $(x^1_t, \dots, x^n) \to (u^1_t, \dots, u^n_t)$ to minimize:

$$\limsup_{T \to \infty} \frac{1}{T} \mathbb{E}^\pi \left[\sum_{t=1}^{T} c(x_t, u_t) \right]$$

- Standard soln requires solving $nd_x \times nd_x$ Riccati Eq.
- Complexity scales $\mathcal{O}(n^3 d_x^3)$.

Outline

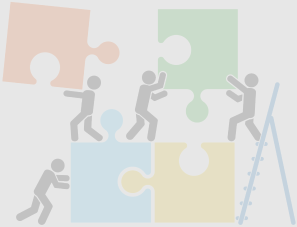


System Model

- Network-coupled subsystems
  - Agents interacting over a graph
  - Coupled dynamics
  - Coupled costs

Planning solution

- Spectral factorization of dynamics and cost
- Decoupled Riccati equations



Learning solution

- Spectral factorization of learning
- Numerical experiments

Our result: Develop a decomposition which computes the optimal policy by solving at most n Riccati eqns of dimension $d_x \times d_x$.

- co-author: Shuang Gao
- paper: `https://arxiv.org/abs/2009.12367`

Spectral decomposition

Spectral decomposition of coupling matrices

$$M = \sum_{\ell=1}^{L} \lambda^{\ell} v^{\ell} (v^{\ell})^{\top},$$

Spectral decomposition

Spectral decomposition of coupling matrices

$$M = \sum_{\ell=1}^{L} \lambda^{\ell} v^{\ell} (v^{\ell})^{\top}, \quad H_q = q_0 I + q_1 \sum_{\ell=1}^{L} \lambda_q^{\ell} v^{\ell} (v^{\ell})^{\top}, \quad H_r = r_0 I + r_1 \sum_{\ell=1}^{L} \lambda_r^{\ell} v^{\ell} (v^{\ell})^{\top}$$

Spectral decomposition

Spectral decomposition of coupling matrices

$$M = \sum_{\ell=1}^{L} \lambda^\ell v^\ell (v^\ell)^\top, \quad H_q = q_0 I + q_1 \sum_{\ell=1}^{L} \lambda_q^\ell v^\ell (v^\ell)^\top, \quad H_r = r_0 I + r_1 \sum_{\ell=1}^{L} \lambda_r^\ell v^\ell (v^\ell)^\top$$

Spectral decomposition of dynamics

At each node $i \in [n]$:

- For each $\ell \in [L]$, define **eigenstates**, **eigencontrols**, and **eigennoise** as

$$x_t^{\ell,i} = x_t^i v^\ell (v^\ell)^\top, \quad u_t^{\ell,i} = u_t^i v^\ell (v^\ell)^\top, \quad \text{and} \quad w_t^{\ell,i} = w_t^i v^\ell (v^\ell)^\top.$$

Spectral decomposition

Spectral decomposition of coupling matrices

$$M = \sum_{\ell=1}^{L} \lambda^\ell v^\ell (v^\ell)^\top, \quad H_q = q_0 I + q_1 \sum_{\ell=1}^{L} \lambda_q^\ell v^\ell (v^\ell)^\top, \quad H_r = r_0 I + r_1 \sum_{\ell=1}^{L} \lambda_r^\ell v^\ell (v^\ell)^\top$$

Spectral decomposition of dynamics

At each node $i \in [n]$:

- For each $\ell \in [L]$, define eigenstates, eigencontrols, and eigennoise as

$$x_t^{\ell,i} = x_t^i v^\ell (v^\ell)^\top, \quad u_t^{\ell,i} = u_t^i v^\ell (v^\ell)^\top, \quad \text{and} \quad w_t^{\ell,i} = w_t^i v^\ell (v^\ell)^\top.$$

- Define auxiliary state, auxiliary control, auxiliary noise as

$$\breve{x}_t^i = x_t^i - \sum_{\ell=1}^{L} x_t^{\ell,i}, \quad \breve{u}_t^i = u_t^i - \sum_{\ell=1}^{L} u_t^{\ell,i}, \quad \text{and} \quad \breve{w}_t^i = w_t^i - \sum_{\ell=1}^{L} w_t^{\ell,i}.$$

Implication of Spectral Decomposition

Noise-coupled dynamics

$$x^{\ell,i}_{t+1} = (A + \lambda^\ell D) x^{\ell,i}_t + (B + \lambda^\ell E) u^{\ell,i}_t + w^{\ell,i}_t$$

and $\breve{x}^i_{t+1} = A \breve{x}^i_t + B \breve{u}^i_t + \breve{w}^i_t$

Implication of Spectral Decomposition

Noise-coupled dynamics

$$x^{\ell,i}_{t+1} = (A + \lambda^{\ell} D) x^{\ell,i}_t + (B + \lambda^{\ell} E) u^{\ell,i}_t + w^{\ell,i}_t$$

and $\breve{x}^i_{t+1} = A\breve{x}^i_t + B\breve{u}^i_t + \breve{w}^i_t$

Decoupled cost

$$c(x_t, u_t) = \sum_{i \in N} \Big[q_0 \breve{c}(\breve{x}^i_t, \breve{u}^i_t) + \sum_{\ell=1}^{L} q^{\ell} c^{\ell}(x^{\ell,i}_t, u^{\ell,i}_t) \Big]$$

where $q^{\ell} = q_0 + q_1 \lambda^{\ell}_q$, $\quad r^{\ell} = r_0 + r_1 \lambda^{\ell}_r$, and

$$\breve{c}(\breve{x}^i_t, \breve{u}^i_t) = (\breve{x}^i_t)^\top Q \breve{x}^i_t + \tfrac{r_0}{q_0} (\breve{u}^i_t)^\top R \breve{u}^i_t$$

$$c^{\ell}(x^{\ell,i}_t, u^{\ell,i}_t) = (x^{\ell,i}_t)^\top Q x^{\ell,i}_t + \tfrac{r^{\ell}}{q^{\ell}} (u^{\ell,i}_t)^\top R u^{\ell,i}_t.$$

Implication of Spectral Decomposition

Eigen-system (ℓ, i) with $\ell \in [L]$, $i \in [n]$

- State $x_t^{\ell,i}$. Control $u_t^{\ell,i}$.
- Dynamics: $x_{t+1}^{\ell,i} = (A + \lambda^\ell D) x_t^{\ell,i} + (B + \lambda^\ell E) u_t^{\ell,i} + w_t^{\ell,i}$
- Per-step cost: $c^\ell(x_t^{\ell,i}, u_t^{\ell,i})$.

Auxiliary system i with $i \in [n]$

- State $\breve{x}_t^i$. Control $\breve{u}_t^i$.
- Dynamics: $\breve{x}_{t+1}^i = A\breve{x}_t^i + B\breve{u}_t^i + \breve{w}_t^i$
- Per-step cost: $c^\ell(\breve{x}_t^i, \breve{u}_t^i)$.

Implication of Spectral Decomposition

Eigen-system (ℓ, i) with $\ell \in [L]$, $i \in [n]$

- State $x_t^{\ell,i}$. Control $u_t^{\ell,i}$.
- Dynamics: $x_{t+1}^{\ell,i} = (A + \lambda^\ell D)x_t^{\ell,i} + (B + \lambda^\ell E)u_t^{\ell,i} + w_t^{\ell,i}$
- Per-step cost: $c^\ell(x_t^{\ell,i}, u_t^{\ell,i})$.

Auxiliary system i with $i \in [n]$

- State $\breve{x}_t^i$. Control $\breve{u}_t^i$.
- Dynamics: $\breve{x}_{t+1}^i = A\breve{x}_t^i + B\breve{u}_t^i + \breve{w}_t^i$
- Per-step cost: $c^\ell(\breve{x}_t^i, \breve{u}_t^i)$.

Only coupled through the noise in the dynamics

Implication of Spectral Decomposition

Eigen-system (ℓ, i) with $\ell \in [L]$, $i \in [n]$

- State $x_t^{\ell,i}$. Control $u_t^{\ell,i}$.
- Dynamics: $x_{t+1}^{\ell,i} = (A + \lambda^\ell D)x_t^{\ell,i} + (B + \lambda^\ell E)u_t^{\ell,i} + w_t^{\ell,i}$
- Per-step cost: $c^\ell(x_t^{\ell,i}, u_t^{\ell,i})$.

Auxiliary system i with $i \in [n]$

- State $\breve{x}_t^i$. Control $\breve{u}_t^i$.
- Dynamics: $\breve{x}_{t+1}^i = A\breve{x}_t^i + B\breve{u}_t^i + \breve{w}_t^i$
- Per-step cost: $c^\ell(\breve{x}_t^i, \breve{u}_t^i)$.

Certainty equivalence: Optimal policy of stochastic LQ system is same as that of deterministic LQ system.

The deterministic system has **decoupled dynamics and cost!**

Only coupled through the noise in the dynamics

Implication of Spectral Decomposition

Eigen-system (ℓ, i) with $\ell \in [L]$, $i \in [n]$

- State $x_t^{\ell,i}$. Control $u_t^{\ell,i}$.
- Dynamics: $x_{t+1}^{\ell,i} = (A + \lambda^\ell D)x_t^{\ell,i} + (B + \lambda^\ell E)u_t^{\ell,i} + \cancel{w_t^{\ell,i}}$
- Per-step cost: $c^\ell(x_t^{\ell,i}, u_t^{\ell,i})$.

Auxiliary system i with $i \in [n]$

- State $\breve{x}_t^i$. Control $\breve{u}_t^i$.
- Dynamics: $\breve{x}_{t+1}^i = A\breve{x}_t^i + B\breve{u}_t^i + \cancel{\breve{w}_t^i}$
- Per-step cost: $c^\ell(\breve{x}_t^i, \breve{u}_t^i)$.

Certainty equivalence: Optimal policy of stochastic LQ system is same as that of deterministic LQ system.

The deterministic system has **decoupled dynamics and cost!**

Only coupled through the noise in the dynamics

Main result

Under standard assumptions, the optimal control action is given by

$$u_t^i = \breve{u}_t^i + \sum_{\ell=1}^{L} u_t^{\ell,i} = \breve{G}\breve{x}_t^i + \sum_{\ell=1}^{L} G^\ell x_t^{\ell,i}$$

where

$$\breve{G} = \mathsf{Gain}\Big(A, B, Q, \frac{r_0}{q_0}R\Big)$$

$$G^\ell = \mathsf{Gain}\Big(A + \lambda^\ell D, B + \lambda^\ell E, Q, \frac{r^\ell}{q^\ell}R\Big)$$

Main result

Under standard assumptions, the optimal control action is given by

$$u_t^i = \breve{u}_t^i + \sum_{\ell=1}^{L} u_t^{\ell,i} = \breve{G}\breve{x}_t^i + \sum_{\ell=1}^{L} G^\ell x_t^{\ell,i}$$

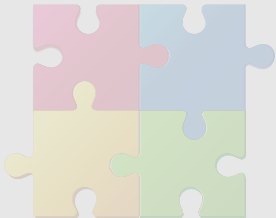
where

$$\breve{G} = \text{Gain}\Big(A, B, Q, \frac{r_0}{q_0} R\Big)$$

$$G^\ell = \text{Gain}\Big(A + \lambda^\ell D, B + \lambda^\ell E, Q, \frac{r^\ell}{q^\ell} R\Big)$$

- The gains $\breve{G}$, $\{G^\ell\}_{\ell=1}^{L}$ are the same at all subsystems!
- Requires solving $(L+1)$ Riccati Eqn of dimension $d_x \times d_x$.
- Complexity scales $\mathcal{O}(L d_x^3)$ (cf. $\mathcal{O}(n^3 d_x^3)$ for naive solution).

Outline

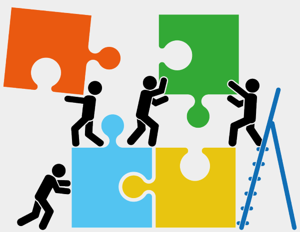


System Model

- Network-coupled subsystems
  - Agents interacting over a graph
  - Coupled dynamics
  - Coupled costs

Planning solution

- Spectral factorization of dynamics and cost
- Decoupled Riccati equations



Learning solution

- Spectral factorization of learning
- Numerical experiments

Review: LQ regulation with unknown/uncertain dynamics

Modeling uncertainty	Model $\theta_\star = [A_\star, B_\star] \in \Theta$ (uncertain set)

Review: LQ regulation with unknown/uncertain dynamics

Modeling uncertainty	Model $\theta_\star = [A_\star, B_\star] \in \Theta$ (uncertain set)

Worst-case design

- Assume that nature is adversarial
- Choose a policy which minimizes worst case performance

- Zero-sum game or robust control

Review: LQ regulation with unknown/uncertain dynamics

Modeling uncertainty

Model $\theta_\star = [A_\star, B_\star] \in \Theta$ (uncertain set)

Worst-case design

- Assume that nature is adversarial
- Choose a policy which minimizes worst case performance

- Zero-sum game or robust control

Learning solution

- Design an adaptive policy which asymptotically converges to the optimal policy of the true (unknown) model.

- Reinforcement learning or adaptive control

Review: LQ regulation with unknown/uncertain dynamics

Modeling uncertainty

Model $\theta_\star = [A_\star, B_\star] \in \Theta$ (uncertain set)

Worst-case design

- Assume that nature is adversarial
- Choose a policy which minimizes worst case performance
- Zero-sum game or robust control

Learning solution

- Design an adaptive policy which asymptotically converges to the optimal policy of the true (unknown) model.
- Reinforcement learning or adaptive control

Comparing learning algorithms

$$\text{Regret}(T) = \sum_{t=1}^{T} \Big[\text{cost of learning algo}(t) - \text{cost of clairvoyant agent}(t)\Big]$$

- Large literature. Various classes of algos with different regret guarantees.

Review: Regret for learning in LQ regulation

Bounds on Regret

- **Lower bound**: No algorithm can do better than $\tilde{\Omega}(d_x^{0.5} d_u \sqrt{T})$.
- **Upper bound**: Various classes of algorithms achieve $\tilde{\mathcal{O}}(d_x^{0.5}(d_x + d_u)\sqrt{T})$.
  - Certainty equivalence; Optimisim in the face of uncertainty; Thompson sampling

Review: Regret for learning in LQ regulation

Bounds on Regret

- **Lower bound**: No algorithm can do better than $\tilde{\Omega}(d_x^{0.5} d_u \sqrt{T})$.
- **Upper bound**: Various classes of algorithms achieve $\tilde{O}(d_x^{0.5}(d_x + d_u)\sqrt{T})$.
  - Certainty equivalence; Optimisim in the face of uncertainty; Thompson sampling

Challenge with learning in networks

- Effective dimensions are $n d_x$ and $n d_u$
- Directly using existing algos gives regret of $\tilde{O}(n^{1.5} d_x^{0.5}(d_x + d_u)\sqrt{T})$.
- Normalized regret per agent is $\tilde{O}(n^{0.5} d_x^{0.5}(d_x + d_u)\sqrt{T})$.

Review: Regret for learning in LQ regulation

Bounds on Regret

- **Lower bound**: No algorithm can do better than $\tilde{\Omega}(d_x^{0.5} d_u \sqrt{T})$.
- **Upper bound**: Various classes of algorithms achieve $\tilde{O}(d_x^{0.5}(d_x + d_u)\sqrt{T})$.
  - Certainty equivalence; Optimisim in the face of uncertainty; Thompson sampling

Challenge with learning in networks

- Effective dimensions are $n d_x$ and $n d_u$
- Directly using existing algos gives regret of $\tilde{O}(n^{1.5} d_x^{0.5}(d_x + d_u)\sqrt{T})$.
- Normalized regret per agent is $\tilde{O}(n^{0.5} d_x^{0.5}(d_x + d_u)\sqrt{T})$.

Regret per agent grows with size of the network!

Our result: Develop a learning algorithm which exploits the structure of the network and has a per agent regret of $\tilde{\mathcal{O}}((1+\frac{1}{n})d_x^{0.5}(d_x+d_u)\sqrt{T})$.

- co-author: Sagar Sudhakara, Ashutosh Nayyar, Yi Ouyang
- paper: https://arxiv.org/abs/2108.07970

Learning model

Problem setting

- **Known:** Network (M, H_q, H_r). Cost (Q, R).
- **Unknown:** Dynamics (A, B, C, D).

Learning model

Problem setting

- **Known:** Network (M, H_q, H_r). Cost (Q, R).
- **Unknown:** Dynamics (A, B, C, D).

Modeling assumptions

- $\breve{\theta}_\star = [A_\star, B_\star] \in \breve{\Theta}$
- $\theta^\ell = [A_\star + \lambda^\ell D_\star, B_\star + \lambda^\ell E_\star] \in \Theta^\ell$
- Bayesian prior on $\breve{\Theta}$ and $\{\Theta^\ell\}_{\ell=1}^L$.

Learning model

Problem setting

- **Known:** Network (M, H_q, H_r). Cost (Q, R).
- **Unknown:** Dynamics (A, B, C, D).

Modeling assumptions

- $\breve{\theta}_\star = [A_\star, B_\star] \in \breve{\Theta}$
- $\theta^\ell = [A_\star + \lambda^\ell D_\star, B_\star + \lambda^\ell E_\star] \in \Theta^\ell$
- Bayesian prior on $\breve{\Theta}$ and $\{\Theta^\ell\}_{\ell=1}^L$.

Implication of Spectral Decomposition

Recall: $c(x_t, u_t) = \sum_{i \in N} \Big[q_0 \breve{c}(\breve{x}_t^i, \breve{u}_t^i) + \sum_{\ell=1}^L q^\ell c^\ell(x_t^{\ell,i}, u_t^{\ell,i}) \Big]$

Thus, for any policy π,

$$J(\pi; \theta_\star) = \sum_{i \in N} \Big[q_0 \breve{J}^i(\pi; \breve{\theta}_\star) + \sum_{\ell=1}^L q^\ell J^{\ell,i}(\pi; \theta_\star^\ell) \Big].$$

Key idea for learning

Separately learn $\{\theta^\ell\}_{\ell=1}^{L}$ and $\breve{\theta}$

- For learning $\theta^\ell_\star$, select an agent $i^\ell_\circ$ such that $\nu^{\ell,i^\ell_\circ} \neq 0$.
- Learn $G^\ell(\theta^\ell_\star)$ using $\{x_t^{\ell,i^\ell_\circ}, u_t^{\ell,i^\ell_\circ}\}_{t\geq 1}$.

Key idea for learning

Separately learn $\{\theta^\ell\}_{\ell=1}^{L}$ and $\breve{\theta}$

- For learning $\theta^\ell_\star$, select an agent $i^\ell_\circ$ such that $\nu^{\ell,i^\ell_\circ} \neq 0$.
- Learn $G^\ell(\theta^\ell_\star)$ using $\{x^{\ell,i^\ell_\circ}_t, u^{\ell,i^\ell_\circ}_t\}_{t\geqslant 1}$.

- At time t, select agent j_{t-1} with the "most informative obs".
- Learn $\breve{G}(\breve{\theta}_\star)$ using $\{\breve{x}^{j_t}_t, \breve{u}^{j_t}_t, \breve{x}^{j_t}_{t+1}\}_{t\geqslant 1}$.

Key idea for learning

Separately learn $\{\theta^\ell\}_{\ell=1}^{L}$ and $\breve{\theta}$

- For learning $\theta^\ell_\star$, select an agent $i^\ell_\circ$ such that $\nu^{\ell, i^\ell_\circ} \neq 0$.
- Learn $G^\ell(\theta^\ell_\star)$ using $\{x^{\ell, i^\ell_\circ}_t, u^{\ell, i^\ell_\circ}_t\}_{t \geqslant 1}$.

- At time t, select agent j_{t-1} with the "most informative obs".
- Learn $\breve{G}(\breve{\theta}_\star)$ using $\{\breve{x}^{j_t}_t, \breve{u}^{j_t}_t, \breve{x}^{j_t}_{t+1}\}_{t \geqslant 1}$.

- Use variant of Thompson sampling to learn each component
- The high-level idea also applies to other learning algos

Key idea for learning

Separately learn $\{\theta^\ell\}_{\ell=1}^L$ and $\breve{\theta}$

- For learning $\theta^\ell_\star$, select an agent $i^\ell_\circ$ such that $\nu^{\ell, i^\ell_\circ} \neq 0$.
- Learn $G^\ell(\theta^\ell_\star)$ using $\{x^{\ell, i^\ell_\circ}_t, u^{\ell, i^\ell_\circ}_t\}_{t \geqslant 1}$.

- At time t, select agent j_{t-1} with the "most informative obs".
- Learn $\breve{G}(\breve{\theta}_\star)$ using $\{\breve{x}^{j_t}_t, \breve{u}^{j_t}_t, \breve{x}^{j_t}_{t+1}\}_{t \geqslant 1}$.

- Use variant of Thompson sampling to learn each component
- The high-level idea also applies to other learning algos

Implication of Spectral Decomposition

Since $J(\pi; \theta_\star) = \sum_{i \in N} \left[q_0 \breve{J}^i(\pi; \breve{\theta}_\star) + \sum_{\ell=1}^L q^\ell J^{\ell, i}(\pi; \theta^\ell_\star) \right]$.

Thus, regret also decomposes as

$$R(T) = \sum_{i \in N} \left[q_0 \breve{R}^i(T) + \sum_{\ell=1}^L q^\ell R^{\ell, i}(T) \right].$$

Bounding regret

Bounding $R^{\ell,i}(T)$

- Since agent $i^\ell_\circ$ is learning in the standard manner, we have
$$R^{\ell,i^\ell_\circ}(T) = \tilde{\mathcal{O}}\big(W^{\ell,i^\ell_\circ} d_x^{0.5}(d_x + d_u)\sqrt{T}\big).$$
- We show that for other agents
$$R^{\ell,i}(T) = \left(\frac{v^{\ell,i}}{v^{\ell,i^\ell_\circ}}\right)^2 R^{\ell,i^\ell_\circ}(T) = \tilde{\mathcal{O}}\big(W^{\ell,i} d_x^{0.5}(d_x + d_u)\sqrt{T}\big).$$

Bounding regret

Bounding $R^{\ell,i}(T)$

- Since agent $i^\ell_\circ$ is learning in the standard manner, we have
$$R^{\ell,i^\ell_\circ}(T) = \tilde{O}\big(W^{\ell,i^\ell_\circ} d_x^{0.5}(d_x + d_u)\sqrt{T}\big).$$
- We show that for other agents
$$R^{\ell,i}(T) = \left(\frac{\nu^{\ell,i}}{\nu^{\ell,i^\ell_\circ}}\right)^2 R^{\ell,i^\ell_\circ}(T) = \tilde{O}\big(W^{\ell,i} d_x^{0.5}(d_x + d_u)\sqrt{T}\big).$$

Bounding $\breve{R}^i(T)$

- Need to bound regret from first principles.
- Using the most informative observation allows us to bound the regret of auxiliary systems at all nodes.
- Show that $\breve{R}^i(T) = \tilde{O}\big(\breve{W}^i d_x^{0.5}(d_x + d_u)\sqrt{T}\big)$.

Bounding regret

Since agent $i_\circ^\ell$ is learning in the standard manner, we have

Bounding $R^{\ell,}$

Overall Regret Bound

- Combining these, we have

  $R(T) = \tilde{\mathcal{O}}\big(\alpha^{\mathcal{G}} d_x^{0.5}(d_x + d_u)\sqrt{T}\big)$, where $\alpha^{\mathcal{G}} = \sum_{\ell=1}^{L} q^\ell + q_0(n - L)$.

- Regret per agent is proportional to

  $$\alpha^{\mathcal{G}}/n = \mathcal{O}\left(1 + \frac{L}{n}\right).$$

  Thus, regret per agent reduces with the size of the network!

the

Bounding $\breve{R}^i$

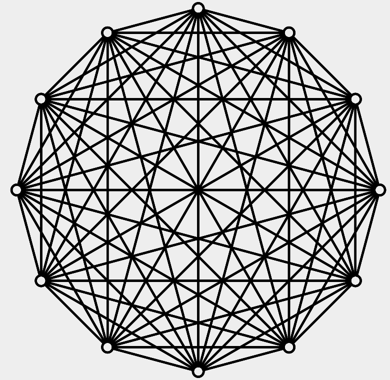
Show that $\breve{R}^i(T) = \tilde{\mathcal{O}}\big(W^i d_x^{0.5}(d_x + d_u)\sqrt{T}\big)$.

Some examples

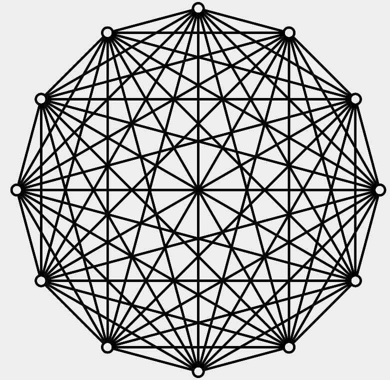
Mean-field systems

Choice of parameters

▷ $M = \frac{1}{n}\mathbb{1}_{n\times n}$ and $H_q = H_r = \frac{1}{n}I + \frac{\kappa}{n}M$.



Mean-field systems



Choice of parameters

- $M = \dfrac{1}{n}\mathbb{1}_{n\times n}$ and $H_q = H_r = \dfrac{1}{n}I + \dfrac{\kappa}{n}M$.

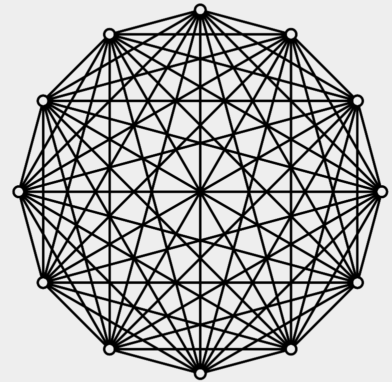
Scaling of regret

- $q^1 = r^1 = (1+\kappa)/n$. Therefore, (normalized) $\alpha^{\mathcal{G}} = \left(1 + \dfrac{\kappa}{n}\right)$.
- Regret per-agent goes down as the network becomes larger (mean-field effect).

Mean-field systems

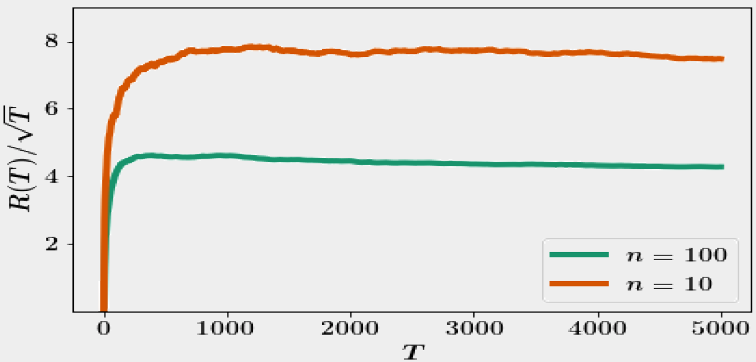
Choice of parameters

- $M = \frac{1}{n}\mathbb{1}_{n\times n}$ and $H_q = H_r = \frac{1}{n}I + \frac{\kappa}{n}M$.

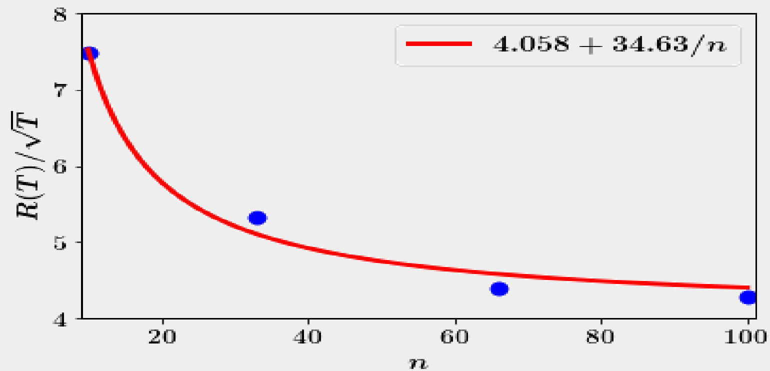


Scaling of regret

- $q^1 = r^1 = (1+\kappa)/n$. Therefore, (normalized) $\alpha^{\mathcal{G}} = \left(1 + \frac{\kappa}{n}\right)$.



$R(T)/\sqrt{T}$ vs T

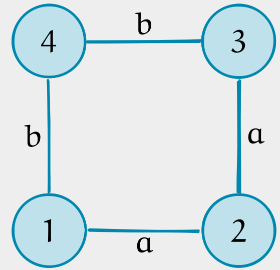


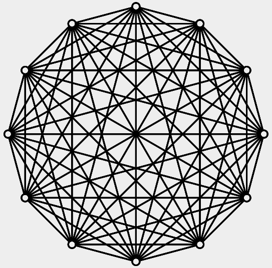
$R(T)/\sqrt{T}$ vs n

A general low-rank network

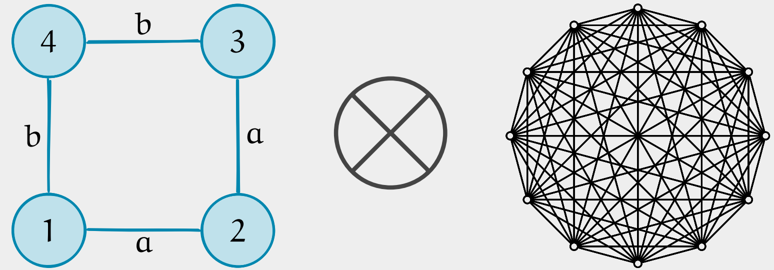
Choice of parameters

- $M = M^\circ \otimes \frac{1}{n}\mathbb{1}_{n\times n}$, $H_q = (I - M)^2$, and $H_r = I$.





A general low-rank network



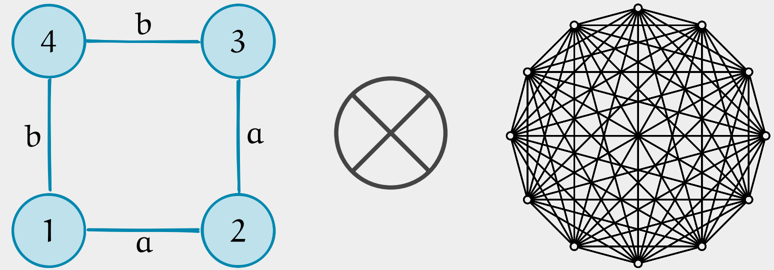
Choice of parameters

- $M = M^\circ \otimes \frac{1}{n}\mathbb{1}_{n\times n}$, $H_q = (I - M)^2$, and $H_r = I$.

Scaling of regret

- $\lambda^\ell = \pm\sqrt{2(a^2 + b^2)}$, $q^\ell = (1 - \lambda^\ell)^2$, $r^\ell = 1$. Therefore, (unnormalized) $\alpha^{\mathcal{G}} = 4n + 4(a^2 + b^2)$.
- Regret per-agent goes down as the network becomes larger (network-field effect).

A general low-rank network

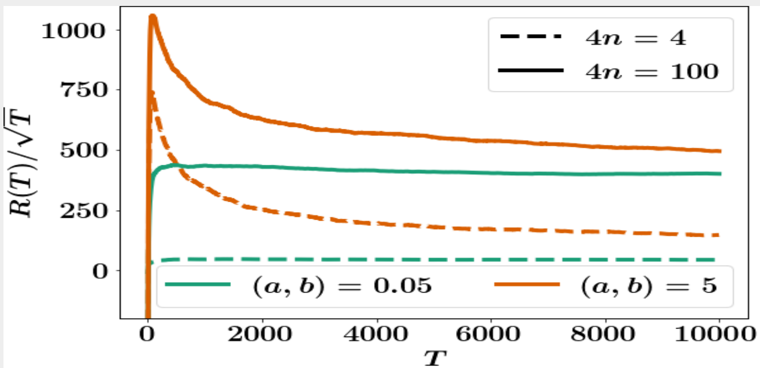


Choice of parameters

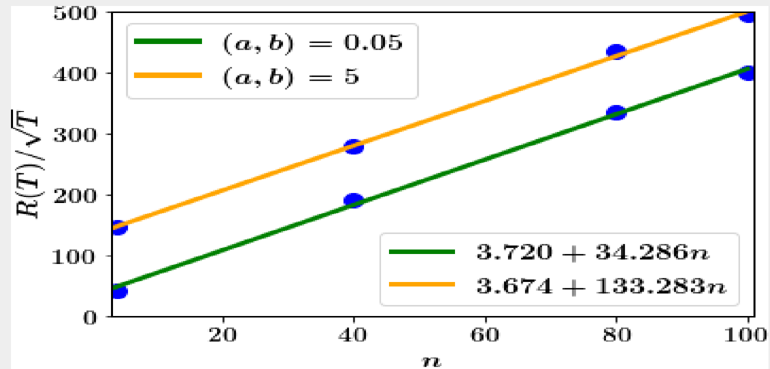
- $M = M^{\circ} \otimes \frac{1}{n}\mathbb{1}_{n \times n}$, $H_q = (I - M)^2$, and $H_r = I$.

Scaling of regret

- $\lambda^{\ell} = \pm\sqrt{2(a^2 + b^2)}$, $q^{\ell} = (1 - \lambda^{\ell})^2$, $r^{\ell} = 1$. Therefore, (unnormalized) $\alpha^{\mathcal{G}} = 4n + 4(a^2 + b^2)$.



$R(T)/\sqrt{T}$ vs T



$R(T)/\sqrt{T}$ vs n

Conclusion

Presented a spectral decomposition method for network-coupled subysstems which leads to scalable planning and learning

Conclusion

Presented a spectral decomposition method for network-coupled subysstems which leads to scalable planning and learning

Planning solution

- Solve $(L+1)$ Riccati eqns of dims $d_x \times d_x$.

Learning solution

- Regret per agent $\tilde{\mathcal{O}}((1+\frac{1}{n})\sqrt{T})$

Conclusion

Presented a spectral decomposition method for network-coupled subysstems which leads to scalable planning and learning

Planning solution

- Solve $(L+1)$ Riccati eqns of dims $d_x \times d_x$.

Learning solution

- Regret per agent $\tilde{\mathcal{O}}((1+\frac{1}{n})\sqrt{T})$

Future Directions

- Multiple types of agents
- Large networks, graphon limits?
- Other types of scalable network stuctures?

- **email**: aditya.mahajan@mcgill.ca
- **web**: http://cim.mcgill.ca/~adityam

Thank you

Funding

- NSERC Discovery
- DND IDEaS Network

References

- **planning**: https://arxiv.org/abs/2009.12367
- **learning**: https://arxiv.org/abs/2108.07970