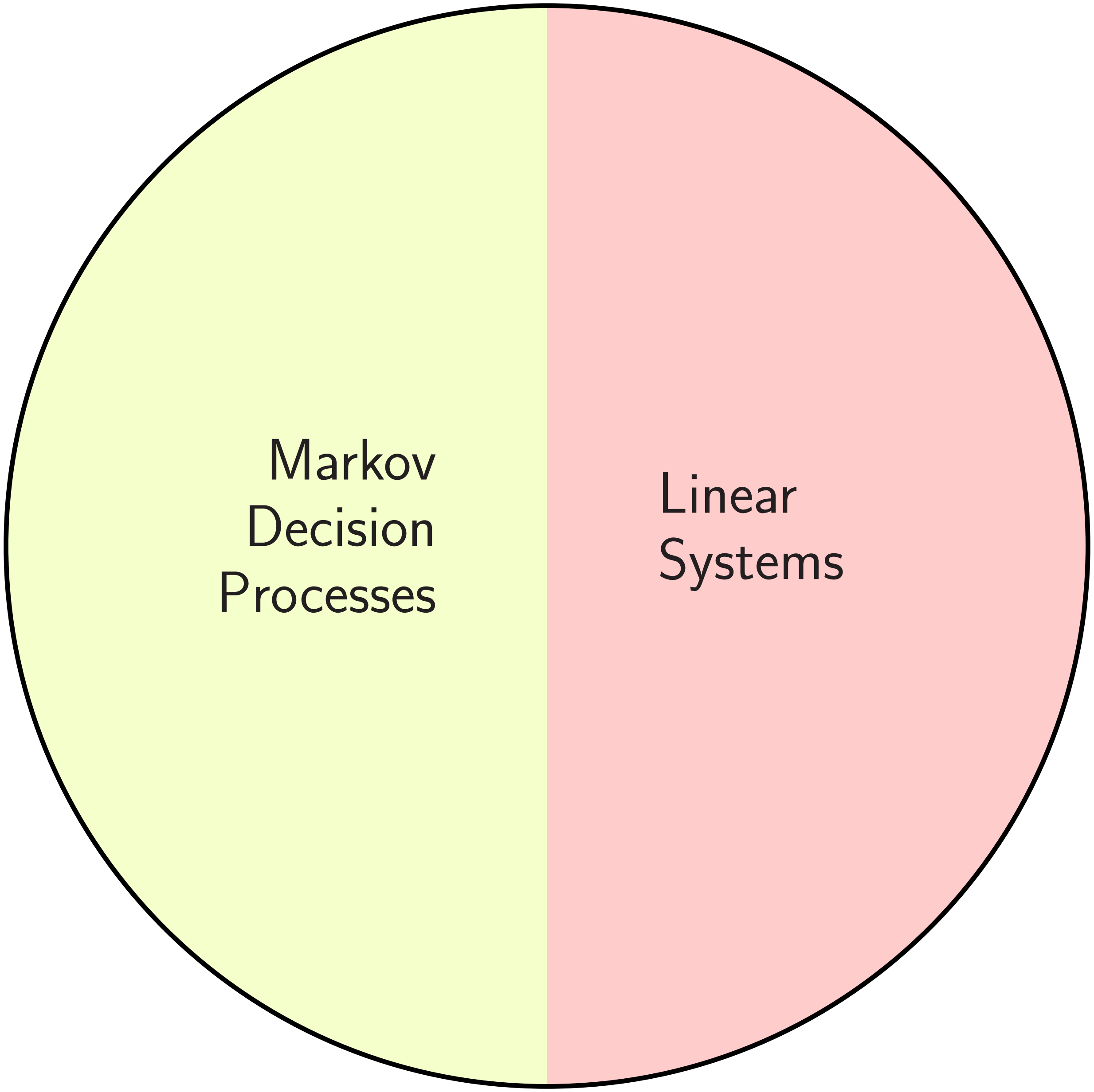
Control on Graphs

Sanjay Lall
Stanford

Joint work with *Sachin Adlakha, Ritesh Madan, Andrea Goldsmith, Randy Cogill*

Workshop on the Frontiers in Distributed Communication, Sensing and Control
October 31 – November 2, 2008
Yale University

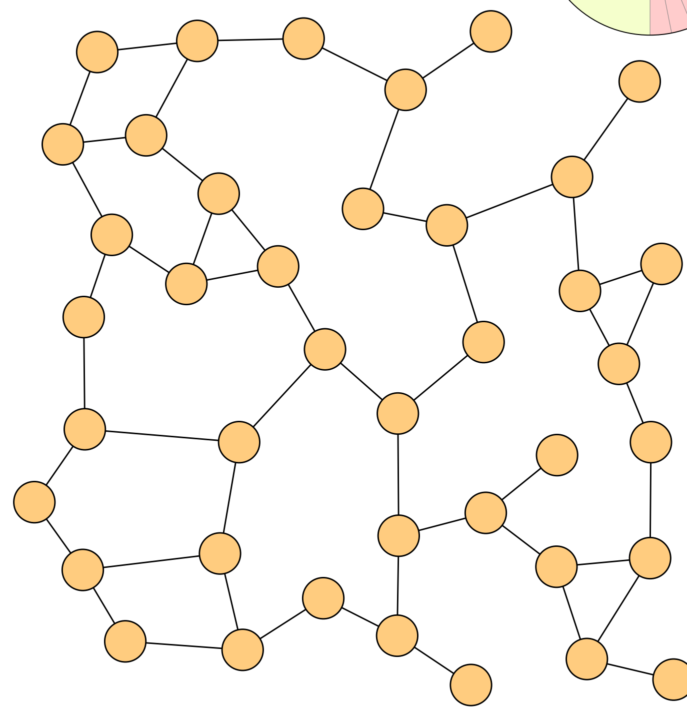
Outline



Linear Systems on Graphs

with Randy Cogill, Systems and Information Engineering, U. Virginia

Interconnected Systems



We know

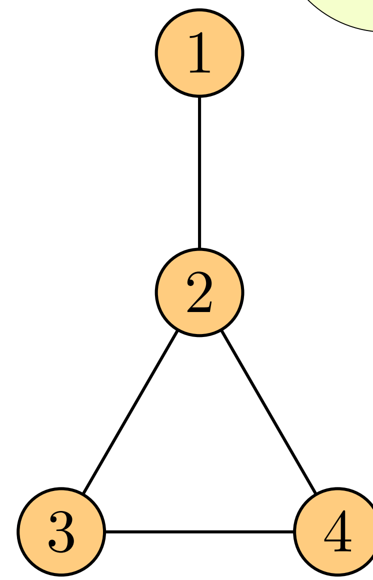
- dynamics of each subsystem
- interaction between neighbors
- an upper bound on the *degree* of the graph

but we *do not* know the number of subsystems

Using only the above, analyze and control the system

Graph

Example: the dynamics associated with this graph are



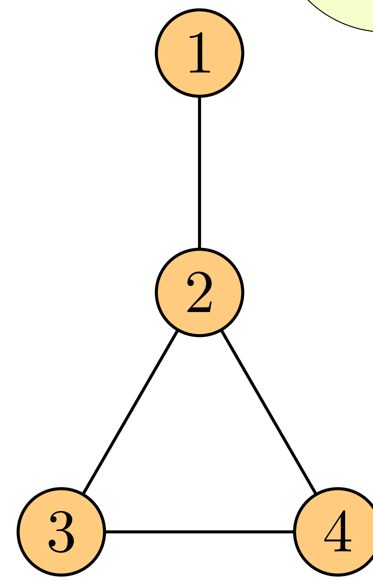
$$
\begin{aligned}
\dot{x}_1 &= Ax_1 + Bx_2 \\
\dot{x}_2 &= Ax_2 + B(x_1 + x_3 + x_4) \\
\dot{x}_3 &= Ax_3 + B(x_2 + x_4) \\
\dot{x}_4 &= Ax_4 + B(x_2 + x_3)
\end{aligned}
$$

In general

$$
\dot{x}_i = Ax_i + B \sum_{j \in \mathcal{N}(i)} x_j
$$

Analysis and Synthesis

$$\dot{x}_i = Ax_i + B \sum_{j \in \mathcal{N}(i)} x_j$$



Given $A, B \in \mathbb{R}^{n \times n}$ and $\deg G \leq d$

- Are the dynamics stable?
- How does the system respond to disturbances?
- Design a decentralized control scheme which is stabilizing
- Minimize the effect of disturbances

Dynamics

The overall dynamics are

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} A & & & B & \\ & A & & & \\ & & \ddots & & B \\ B & & & A & \\ & & B & & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_n \end{bmatrix}$$

Call the above matrix $A_{\mathsf{sys}}(G)$.

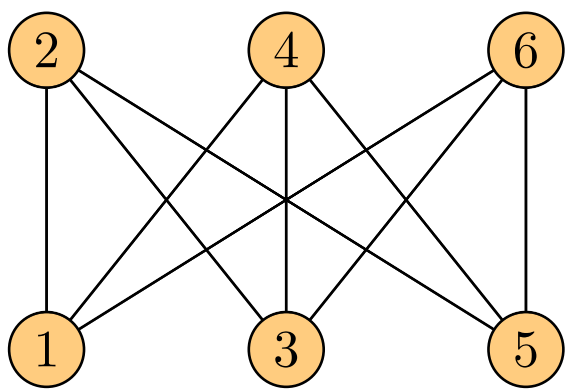
- Sparsity pattern determined by G
- If $\deg G \leq d$ then at most d off-diagonal blocks are nonzero in each row

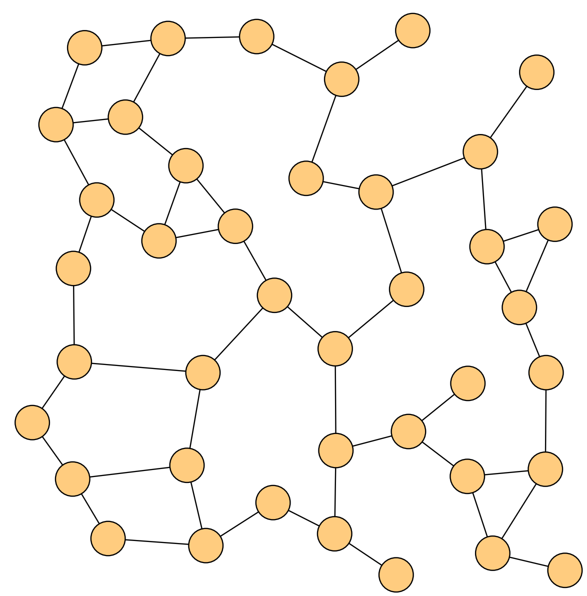
Stability

Theorem. The system $A_{\text{sys}}(K_{d,d})$ is strongly stable if and only if

$$A_{\text{sys}}(G) \text{ is strongly stable for all } G \text{ such that } \deg G \leq d$$

e.g. strong stability of $K_{3,3}$ is equivalent to strong stability of *all graphs with deg* ≤ 3.





Reduced to graph with $2d$ *vertices, scales to arbitrary number of vertices*

Strong Stability

A system is called *strongly stable* if it has a Lyapunov function of the form

$$V(x) = \sum_{i=1}^{n} x_i^T Q_i x_i$$

Equivalent to an SDP. A is strongly stable iff there exists a block diagonal matrix Q such that

$$AQ + QA^T < 0 \quad \text{and} \quad Q > 0$$

Scalability

The above result scales with degree d, but not with the number of nodes

The result below eliminates the dependence on d.

Theorem. The following are equivalent:

(i) $\begin{bmatrix} A & dB \\ dB & A \end{bmatrix}$ is strongly stable

(ii) $A_{\mathsf{sys}}(G)$ is strongly stable for all G such that $\deg G \leq d$

- Reduction to a graph with 2 vertices!
- But scale the interconnection by d

SDP Condition

The following are equivalent:

(i) There exists a matrix $Q > 0$ such that

$$\begin{bmatrix} A & dB \\ dB & A \end{bmatrix} \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} + \begin{bmatrix} Q & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} A & dB \\ dB & A \end{bmatrix}^T < 0$$

(ii) $A_{\mathsf{sys}}(G)$ is strongly stable for all G such that $\deg G \leq d$

Main Idea

Suppose P and Q are symmetric, then

$$\begin{bmatrix} P & dQ \\ dQ & P \end{bmatrix} > 0$$

if and only if

$$Z_{\text{graph}}(G) > 0 \text{ for all } G \text{ with } \deg G \leq d$$

Here Z_{graph} is defined to have the sparsity pattern of G.

$$Z_{\text{graph}}(G) = \begin{bmatrix} P & & & Q & \\ & P & & & \\ & & \ddots & & Q \\ Q & & & P & \\ & & Q & & P \end{bmatrix}$$

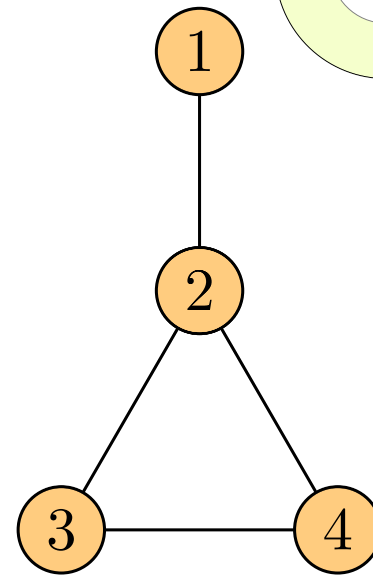
Synthesis

Add *control input* u and *disturbance* w.

$$\dot{x}_1 = Ax_1 + Bx_2 + Cu_1 + Dw_1$$

$$\vdots$$

$$\dot{x}_4 = Ax_4 + B(x_2 + x_3) + Cu_4 + Dw_4$$



Use controller $u_i = Kx_i$, and pick K to minimize $\mathrm{E}\|x\|^2$

- Apply convex synthesis technique to design K for the bipartite graph
- Use a block-diagonal storage function
- Bound on response for all graphs with $\deg G \le d$

Decentralized Stabilization

Find

$$K = \begin{bmatrix} K_1 & & & \\ & K_2 & & \\ & & \ddots & \\ & & & K_n \end{bmatrix}$$

such that

$$A + CK \text{ is Hurwitz}$$

- Called *decentralized* stabilization
- Each controller knows only its own state
- Much more difficult than simple stabilization

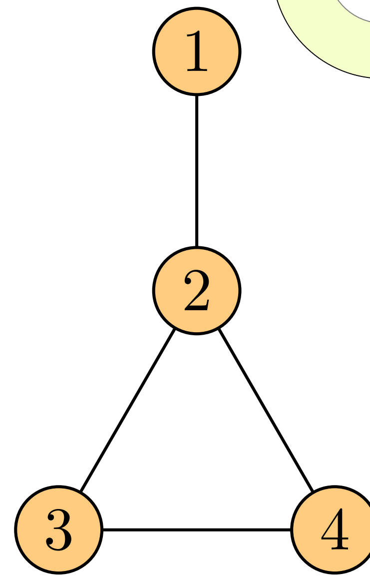
Synthesis

With *control input* u

$$\dot{x}_1 = Ax_1 + Bx_2 + Cu_1$$

$$\vdots$$

$$\dot{x}_4 = Ax_4 + B(x_2 + x_3) + Cu_4$$



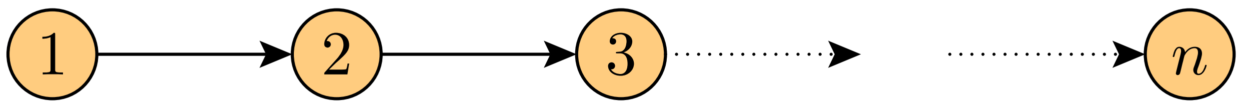
- Solve an SDP to find K such that

$$\begin{bmatrix} A + CK & dB \\ dB & A + CK \end{bmatrix} \quad \text{is strongly stable}$$

- The controller $u_i = Kx_i$ stabilizes all graphs with deg $\leq d$,

Example: Uncertainty Propagation

Stable system with 3 states. Every chain is stable



But small forcing is amplified dramatically as we increase chain length

n	5	10	20	30
norm	11.8	222.4	7.7×10^4	2.6×10^7

We design a decentralized controller such that

- Every system with $\deg G \leq 3$ has norm ≤ 1.82.
- For the above chains we achieve a norm ≤ 0.34 for all the above n.

Directed Graphs

Theorem. The following are equivalent:

(i) $\begin{bmatrix} A & dB \\ dB & A \end{bmatrix}$ and $\begin{bmatrix} A & 0 \\ dB & A \end{bmatrix}$ are strongly stable,

certifiable with the same Lyapunov function

(ii) $A_{\mathsf{sys}}(G)$ is strongly stable for all G such that $\deg G \leq d$

- Above stability analysis is equivalent to an SDP
- Extends to synthesis using standard change of variables

Summary: Linear Systems on Graphs

Strong stability over all graphs with $\deg(G) \leq d$
is equivalent to an SDP

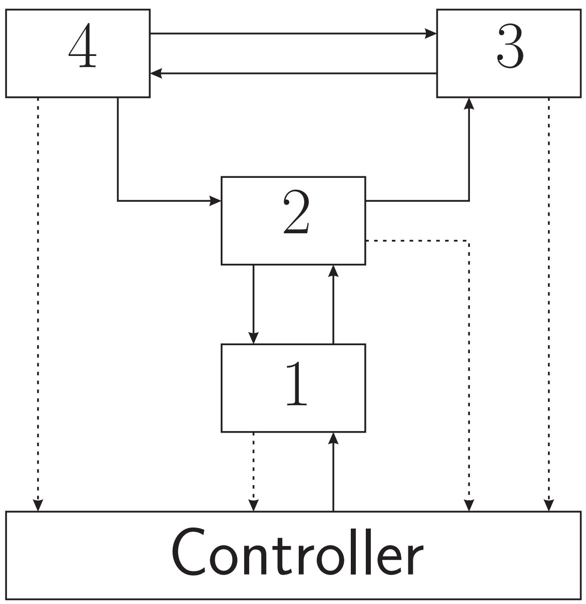
- undirected graphs
- directed graphs
- non-identical subsystems

MDPs on Graphs

with Sachin Adlakha, Ritesh Madan, Andrea Goldsmith

MDPs on Graphs

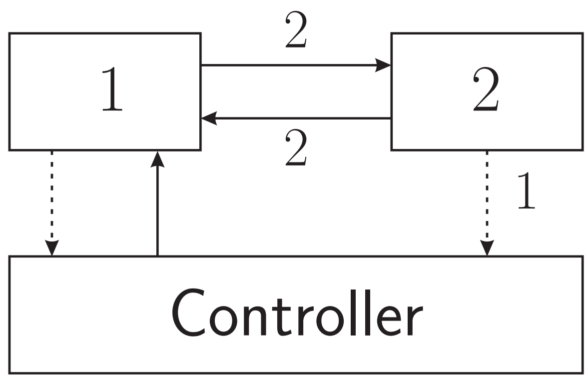
- Markov Decision Processes (MDPs) connected by delay links
- Partially-Observed (POMDP) formulation hard



An optimal controller exists with *finite memory*

Example: Two Subsystems

$$x^1_{t+1} = f^1\left(x^1_t, u_t, w^1_t, x^2_{t-2}\right)$$
$$x^2_{t+1} = f^2\left(x^2_t, w^2_t, x^1_{t-2}\right)$$



The optimal controller has the form

$$u_t = \mu_t\Big(x^1_t,\ x^1_{t-1},\ x^1_{t-2},\ x^1_{t-3},\ x^2_{t-1},\ x^2_{t-2}\Big)$$

Example: Delayed Measurement

- Single system with delayed state availability.
- The system dynamics are

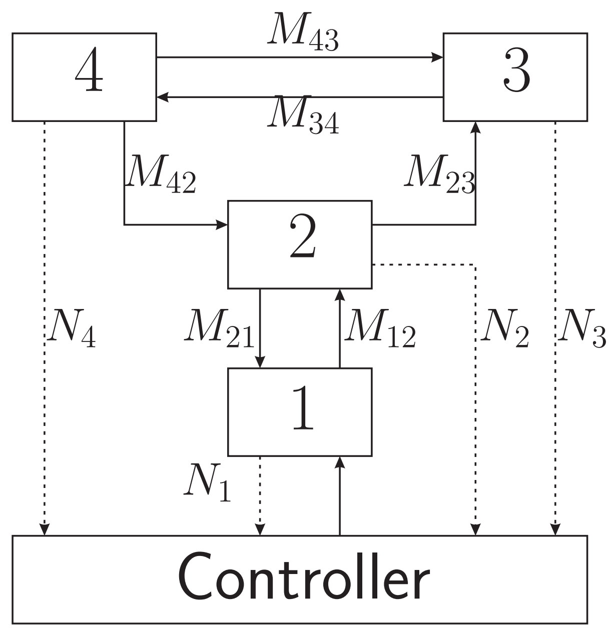
$$x_{t+1} = f\left(x_t, u_t, w_t\right)$$

- At time t, the controller has access to $u_0, \ldots, u_{t-1}$ and $x_0, \ldots, x_{t-N}$.

There is an optimal controller of the form

$$u_t = \mu_t(u_{t-N}, \ldots, u_{t-1},\ x_{t-N})$$

General Model



Dynamics have the form:

$$x_{t+1}^2 = f^2\left(x_t^2, w_t^2, x_{t-M_{12}}^1, x_{t-M_{42}}^4\right)$$

Controller chooses u_t based on $x_1(t-N_1), x_2(t-N_2), \ldots$

Main Result

There exists an optimal controller of the form $u_t = \mu_t^{\mathsf{opt}}(y_t^{\mathsf{mem}})$ where

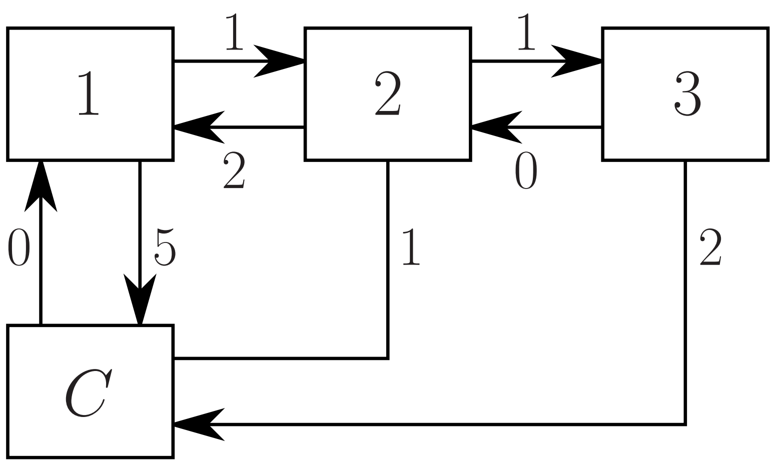
$$y_t^{\mathsf{mem}} = \left(u^i_{t-d_i}, \ldots, u^i_{t-1},\; x^i_{t-N_i-b_i} \ldots, x^i_{t-N_i}\right)$$

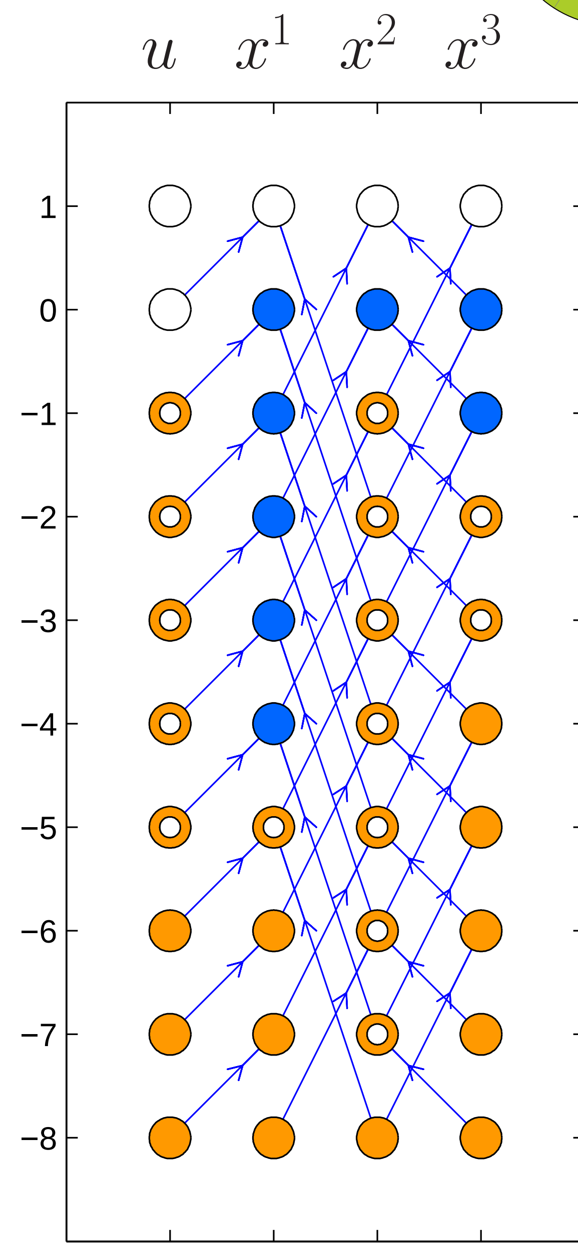
Optimal controller only needs *finite part* of the observation history

$$b_i = \max\left\{d_i, d_s + M_{is} \mid s \in \mathsf{ch}(i)\right\} - N_i$$

$$d_i = \max\left\{N_i, N_s - 1 - M_{si} \mid s \in \mathsf{pa}(i)\right\}$$

Markov Blanket





- Extensive form graph
- Blue vertices unmeasurable
- Belief state only depends on Markov blanket of blue nodes
- Inductive proof; future u_t may depend on earlier states

Dynamic Programming proof

Over a finite time interval, we have cost function

$$\mathrm{E}\sum_{t=0}^{N} g_t(x_t, u_t)$$

- For POMDPs the value function is

$$V_t(y_0, \ldots, y_t, u_0, \ldots, u_{t-1}) \\ = \mathrm{E}(g_t + \cdots + g_N \mid y_0, \ldots, y_t, u_0, \ldots, u_{t-1})$$

- We show, by induction that there exists $\hat{V}_t$ such that

$$V_t(y_0, \ldots, y_t, u_0, \ldots, u_{t-1}) = \hat{V}_t(y_t^{\mathsf{mem}})$$

Information State & Suff. Statistics

For more general, infinite-time problems, we show

y_t^{mem} is an information state

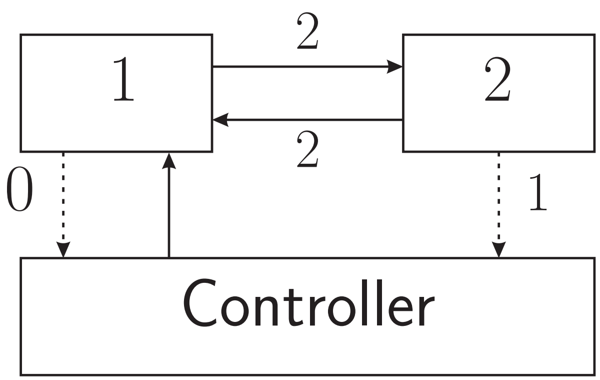
allowing construction of the optimal controller via DP with y_t^{mem} as state.

ξ_t, a function of $u_0, \ldots, u_{t-1}, y_0, \ldots, y_t$ is called an *information state* if

- $\operatorname{Prob}\left(\xi_{t+1} \mid \xi_0, \ldots, \xi_t, u_0, \ldots, u_t\right)$ is independent of the policy
- $\operatorname{E}\left(g_t\left(x_t, a_t\right) \mid \xi_t, u_t\right)$ is independent of the policy
- With probabilities independent of the policy:

$$\operatorname{Prob}\Big(x_t \mid \xi_0, \ldots, \xi_t\Big) = \operatorname{Prob}\left(x_t \mid y_0, \ldots, y_t, u_0, \ldots, u_{t-1}\right).$$

Example



The optimal controller is

$$u_t = a_0 x_t^1 + a_1 x_{t-1}^1 + a_2 x_{t-2}^1 + a_3 x_{t-3}^1 + b_0 x_{t-1}^2 + b_1 x_{t-2}^2,$$

where,

$$a_0 = -0.83, \quad a_1 = -0.13, \quad a_2 = -0.13 \quad a_3 = -0.14$$
$$b_0 = -0.64, \quad b_1 = -0.17$$

Summary: MDPs on Graphs

An optimal controller exists with finite memory.

- Centralized control, distributed delayed state feedback
- Bandwidths depend only on the graph structure and the delays.
- Gives tractable algorithm for exact solution of a class of POMDPs.
- Decentralized linear systems?