Structure of optimal decentralized control policies
An axiomatic approach

Aditya Mahajan
Yale

Joint work with: Demos Teneketzis (UofM), Sekhar Tatikonda (Yale), Ashutosh Nayyar (UofM), Serdar Yüksel (Queens)

March 1, 2010, Notre Dame

Decentralized systems are everywhere . . .

Examples of decentralized systems

Communication Systems

- Wireless networks
- Cognitive radios
- Multimedia communication
- Scheduling and routing in Internet
- Social networks

Surveillance and Sensor Nets

- Disaster monitoring
- Calibration and validation of remote sensing observations
- Fleet of unmanned aerial vehicles
- Intruder detection in networks

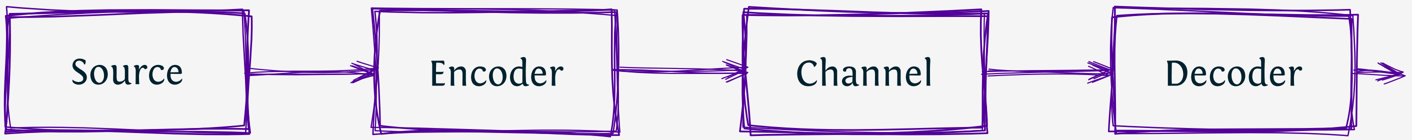
Networked control sys

- Manufacturing plants
- Transportation networks
- Real-time route scheduling
- Aerospace applications

And many more . . .

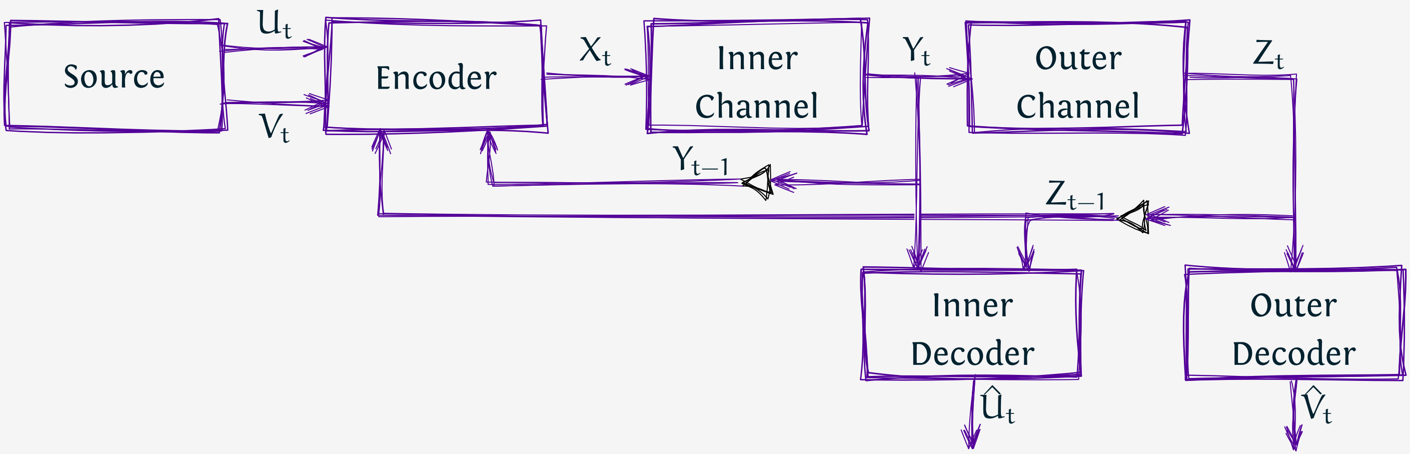
- Coordination in robotics
- On-time diagnosis in nuclear power plants
- Fault monitoring in power grids
- Task scheduling in multi-core CPUs

Real-time communication



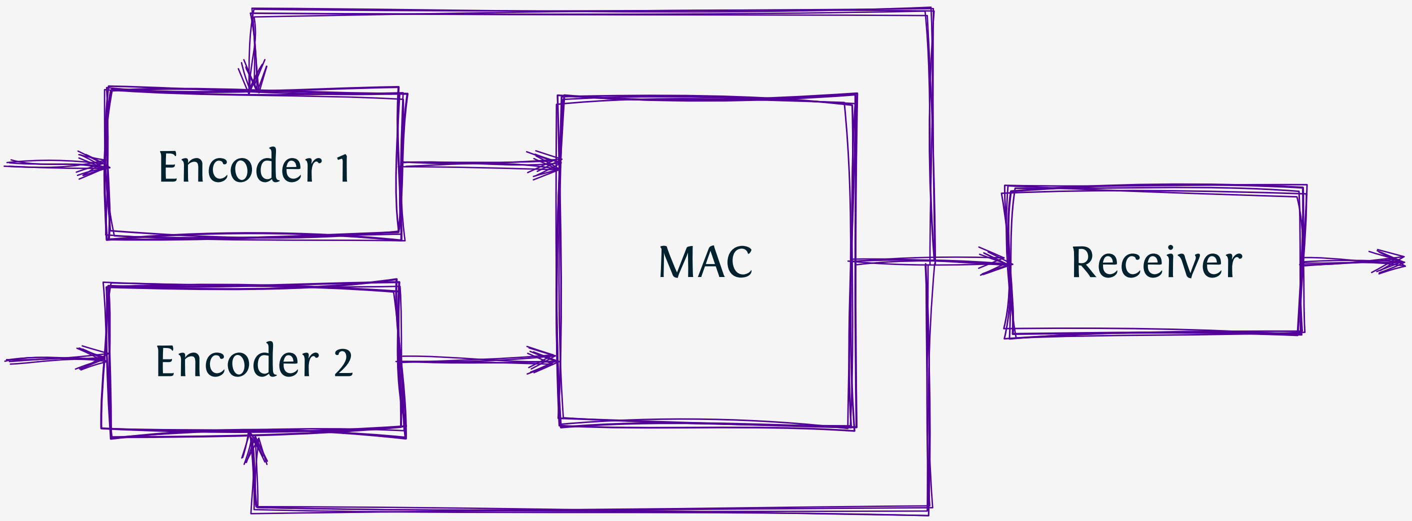
M-Teneketzis, TIT 09

Broadcast with feedback



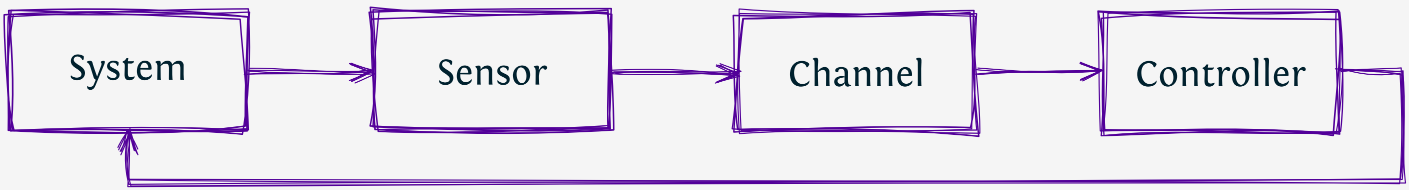
M, Allerton 09

MAC with feedback



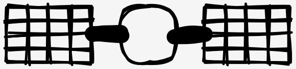
M, ITA 10

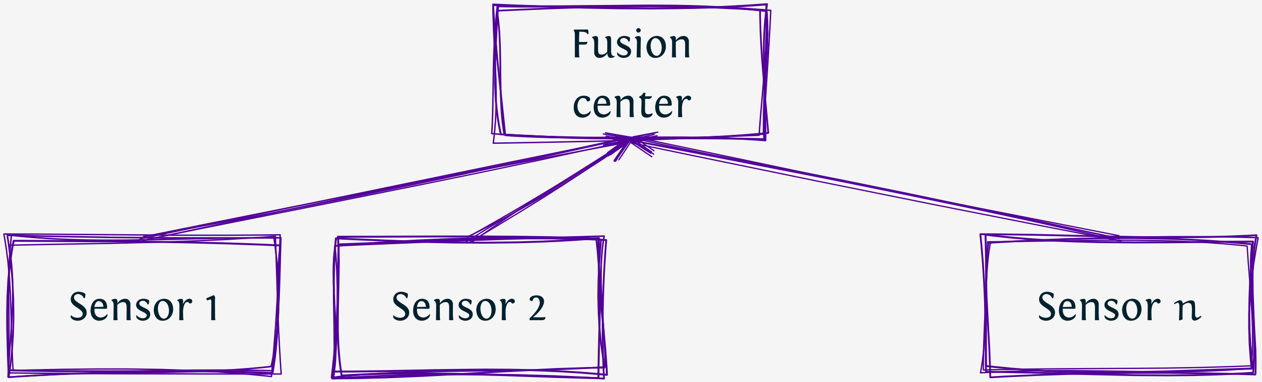
Control over noisy channels



M-Teneketzis, SICON 09

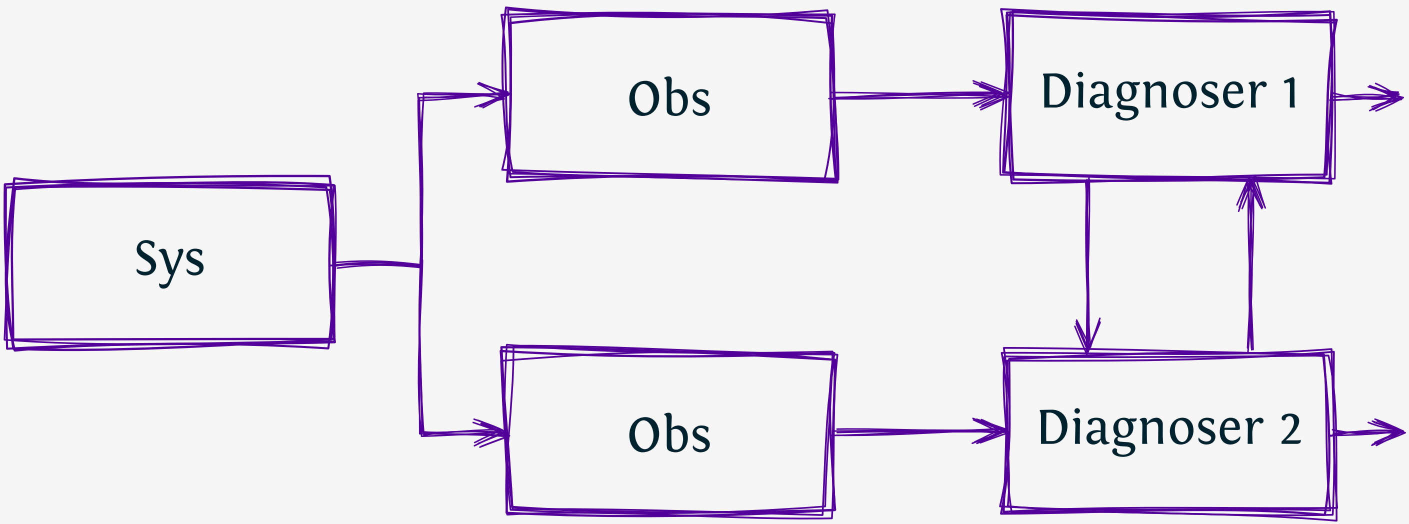
Calibration and validation of remote sensing





Shuman-Nayyar-M-*et al.* Proc IEEE, 10, JSTARS 10

On-time diagnosis with communication



Basic research premise

- The various applications where decentralized systems arise are **independent areas of research** with dedicated communities.

Basic research premise

- The various applications where decentralized systems arise are **independent areas of research** with dedicated communities.
- Nonetheless, these applications share **common features** and **common design principles**.

Basic research premise

- The various applications where decentralized systems arise are **independent areas of research** with dedicated communities.
- Nonetheless, these applications share **common features** and **common design principles**.
-

Develop a **systematic methodology**
that addresses these commonalities.

Basic research premise

- The various applications where decentralized systems arise are **independent areas of research** with dedicated communities.
- Nonetheless, these applications share **common features** and **common design principles**.
-

> Develop a **systematic methodology**
> that addresses these commonalities.

- Such a methodology will provide **design guidelines** for all applications.

Systematic design of decentralized systems

Structure of optimal policies

The data at the controllers increases with time, leading to a doubly exponential increase in the number of policies.

When can an agent, or a group of agents,

- shed available information
- compress available information

without loss of optimality?

Systematic design of decentralized systems

Structure of optimal policies

The data at the controllers increases with time, leading to a doubly exponential increase in the number of policies.

When can an agent, or a group of agents,

- shed available information
- compress available information

without loss of optimality?

Search of optimal policies

- Brute force search of an optimal policy has doubly exponential complexity with time-horizon.
- How can we search for an optimal policy efficiently?
- How can we implement an optimal policy efficiently?

Systematic design of decentralized systems

Structure of optimal policies

The data at the controllers increases with time, leading to a doubly exponential increase in the number of policies.

When can an agent, or a group of agents,

- shed available information
- compress available information

without loss of optimality?

Search of optimal policies

- Brute force search of an optimal policy has doubly exponential complexity with time-horizon.
- How can we search for an optimal policy efficiently?
- How can we implement an optimal policy efficiently?

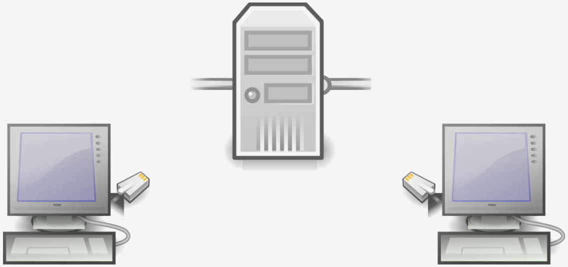
Design principles

- Can we check if the optimal design of a decentralized system is tractable, **without actually designing the system**?
- Can we provide additional information to agents to make the design tractable? If so, can we find the smallest such information?

Outline

1. Why are decentralized systems difficult: an example
2. Overview of decentralized systems
   - Classification
   - Literature overview
3. Overview of centralized stochastic control
4. Systematic derivation of structural properties
   - Shed irrelevant information
   - Compress common information
5. Automated derivation using graphical models
6. Conclusion

Multiaccess broadcast



Multiaccess broadcast

MAB Channel

- Single user transmits $\Rightarrow$ 
- both users transmit $\Rightarrow$ 

Transmitters

- Packet arrival is independent Bernoulli process
- Queues with buffer of size 1
- Packet held in queue until successful transmission

Channel feedback

- A user knows whether its transmission was successful or not.

Multiaccess broadcast

Policy of transmitter

$$U^i_t = g_{i,t}(X^i_{1:t}, U^i_{1:t-1}, Z_{1:t-1})$$

MAB Channel

- Single user transmits $\Rightarrow$ 
- both users transmit $\Rightarrow$ 

Transmitters

- Packet arrival is independent Bernoulli process
- Queues with buffer of size 1
- Packet held in queue until successful transmission

Channel feedback

- A user knows whether its transmission was successful or not.

Multiaccess broadcast

Policy of transmitter

$$U_t^i = g_{i,t}(X_{1:t}^i, U_{1:t-1}^i, Z_{1:t-1})$$

Objective

Maximize throughput or minimize delay

- Avoid collisions
- Avoid idle

MAB Channel

- Single user transmits $\Rightarrow$ 
- both users transmit $\Rightarrow$ 

Transmitters

- Packet arrival is independent Bernoulli process
- Queues with buffer of size 1
- Packet held in queue until successful transmission

Channel feedback

- A user knows whether its transmission was successful or not.

History of multiaccess broadcast

Hluchyj and Gallager, NTC 81

- Considered symmetric arrival rates
- Restricted attention to "window protocols"

History of multiaccess broadcast

Hluchyj and Gallager, NTC 81

- Considered symmetric arrival rates
- Restricted attention to "window protocols"

Ooi and Wornell, CDC 96

- Considered a relaxation of the problem
- Numerically find optimal policy of the relaxed problem
- Hluchyj and Gallager's scheme meets this upper bound

History of multiaccess broadcast

Hluchyj and Gallager, NTC 81

- Considered symmetric arrival rates
- Restricted attention to "window protocols"

Ooi and Wornell, CDC 96

- Considered a relaxation of the problem
- Numerically find optimal policy of the relaxed problem
- Hluchyj and Gallager's scheme meets this upper bound

Recent AI literature

- One of the benchmark problems for decentralized systems
- Consider the case of asymmetric arrival rates
- Approximate heuristic solutions for small horizons

Design Questions

- **Difficulty:** Data at the controllers increases with time.
  - Number of control policies increases doubly exponentially with time, making search for optimal policy difficult.
  - Difficult to implement control functions with time increasing domain

Design Questions

- **Difficulty:** Data at the controllers increases with time.
  - Number of control policies increases doubly exponentially with time, making search for optimal policy difficult.
  - Difficult to implement control functions with time increasing domain
- Hluchyj and Gallager circumvented the difficulty by restricting attention to “window protocols”.

Design Questions

- **Difficulty:** Data at the controllers increases with time.
  - Number of control policies increases doubly exponentially with time, making search for optimal policy difficult.
  - Difficult to implement control functions with time increasing domain
- Hluchyj and Gallager circumvented the difficulty by restricting attention to "window protocols".
- Is such a restriction optimal?

Ooi and Wornell consider a relaxed problem whose optimal solution (found numerically!) happens to be identical to the strategy of Hluchyj and Gallager.

Design Questions

- **Difficulty:** Data at the controllers increases with time.

Even simple problems remain unresolved for decades

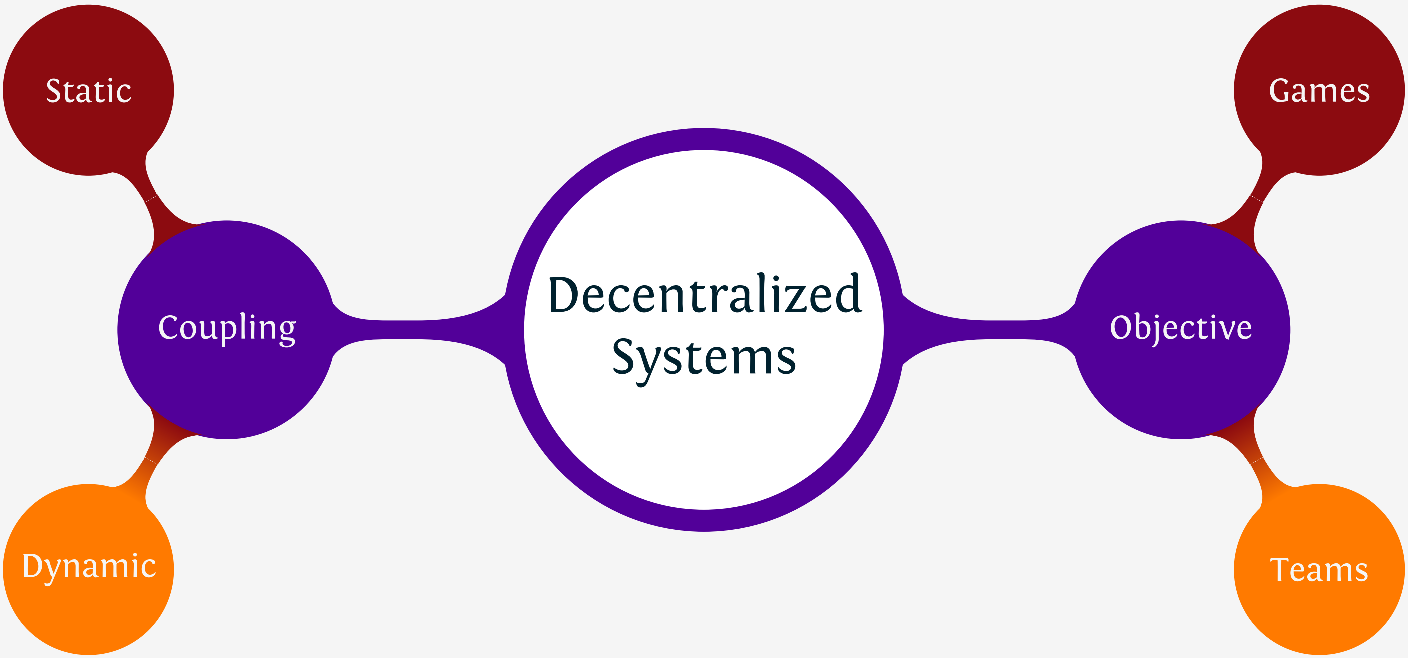
- Is such a restriction optimal?

Ooi and Wornell consider a relaxed problem whose optimal solution (found numerically!) happens to be identical to the strategy of Hluchyj and Gallager.

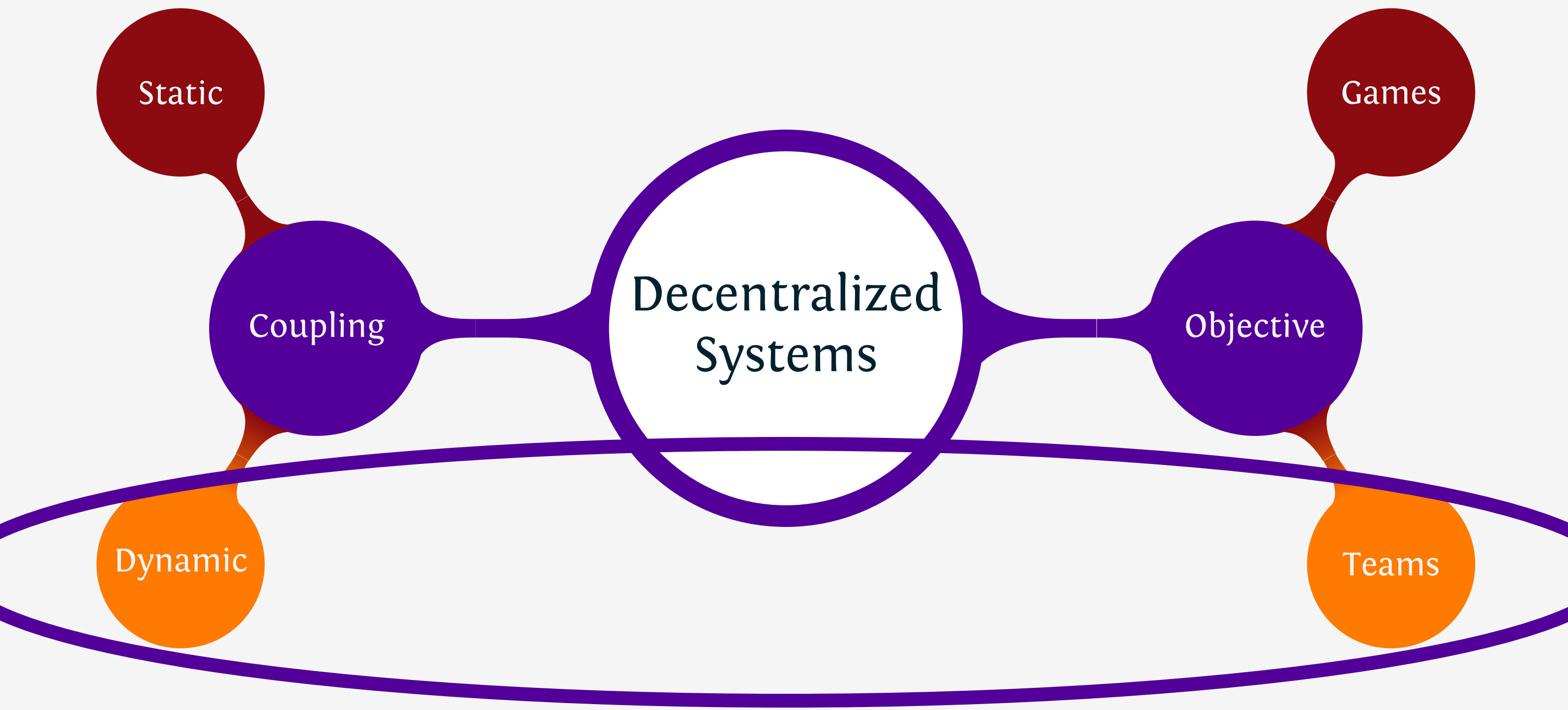
Outline

1. Why are decentralized systems difficult: an example
2. **Overview of decentralized systems**
   - **Classification**
   - **Literature overview**
3. Overview of centralized stochastic control
4. Systematic derivation of structural properties
   - Shed irrelevant information
   - Compress common information
5. Automated derivation using graphical models
6. Conclusion

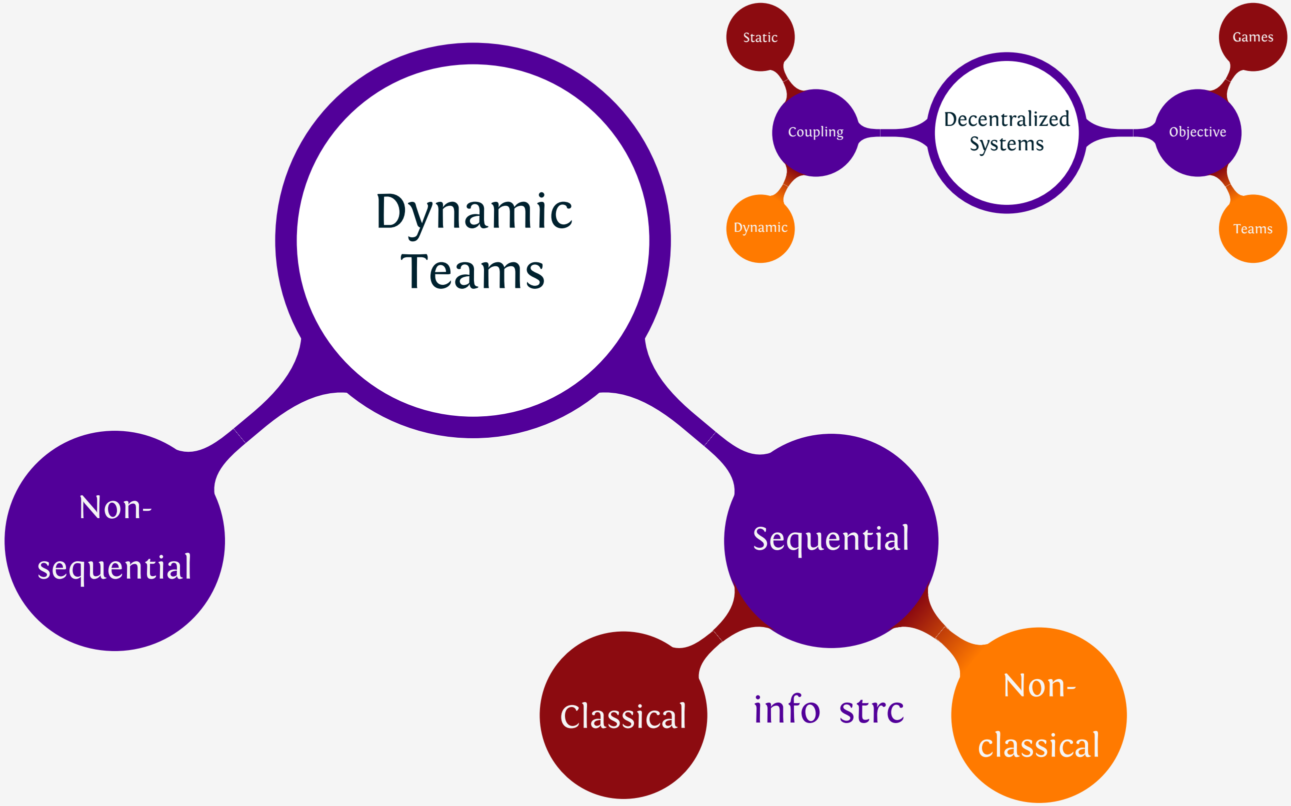
Classification of decentralized systems



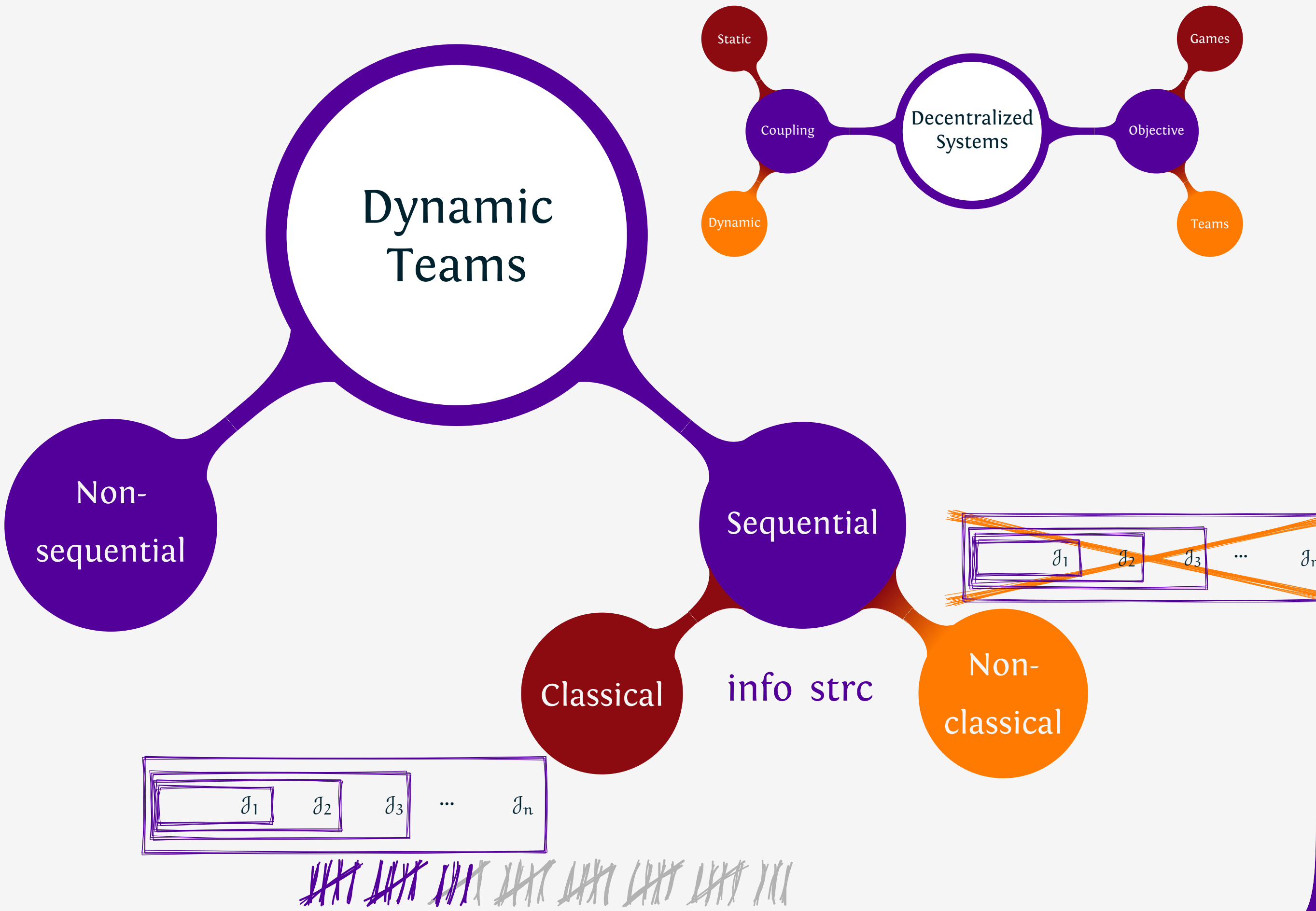
Classification of decentralized systems



Classification of decentralized systems

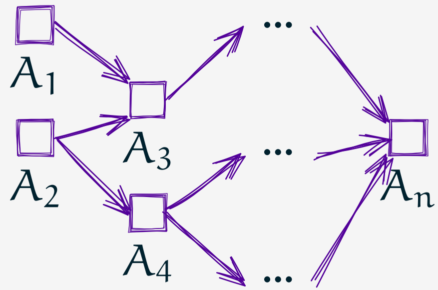


Classification of decentralized systems

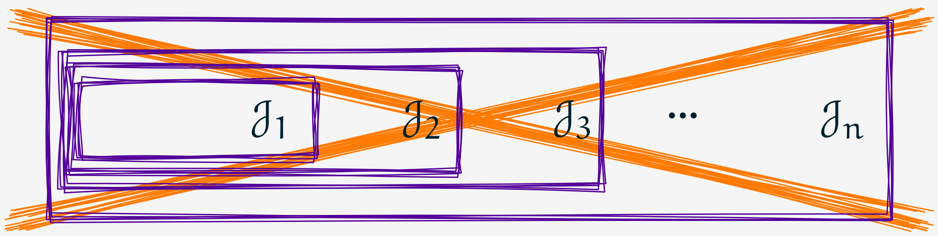


We are interested in

Sequential dynamic teams



with non-classical information structures



Literature Overview : Economics

TEAM DECISION PROBLEMS[1]

By R. Radner

University of California, Berkeley

1. Introduction. In a *team decision problem* there are two or more decision variables, and these different decisions can be made to depend upon different aspects of the environment, i.e., upon different information variables. For ex-

ECONOMIC THEORY OF TEAMS

by
JACOB MARSCHAK and ROY RADNER

Yale University Press, New Haven and London
1972

Literature Overview : Controls

SIAM J. CONTROL
Vol. 9, No. 2, May 1971

ON INFORMATION STRUCTURES, FEEDBACK AND CAUSALITY*

H. S. WITSENHAUSEN†

Abstract. A finite number of decisions, indexed by $\alpha \in A$, are to be taken. Each decision amounts to selecting a point in a measurable space $(U_\alpha, \mathscr{F}_\alpha)$. Each decision is based on some information fed back from the system and characterized by a subfield $\mathscr{I}_\alpha$ of the product space $(\prod_\alpha U_\alpha, \prod_\alpha \mathscr{F}_\alpha)$. The decision function for each α can be any function γ_α measurable from $\mathscr{I}_\alpha$ to $\mathscr{F}_\alpha$.

PROCEEDINGS OF THE IEEE, VOL. 59, NO. 11, NOVEMBER 1971

Separation of Estimation and Control for Discrete Time Systems

HANS S. WITSENHAUSEN, MEMBER, IEEE

Invited Paper

Literature Review : Negative results

- H.S. Witsenhausen, *counterexample in stochastic control*, SICON 1968

Linear policies are not optimal for linear quadratic Gaussian systems under non-classical information structure

- D.S. Bernstein, S. Zilberstein, and N. Immerman, 2000

In general, the problem is NEXP-complete: **no polynomial time** solution can exist.

Literature Review : Few general results

- Standard form: Witsenhausen 1973
- Non-classical LQG problems: Sandell and Athans, 1974
- Multi-criteria problems: Basar, 1978
- Equivalence of static and dynamic teams: Witsenhausen 1988
- Non-sequential systems: Andersland and Teneketzis, 1992 and 1994.
- Two agent teams: M, 2008.

Literature Review: Specific info structures

- Partially nested info structures, Ho and Chu, 1972, Ho, Kastner, and Wong, 1978, Ho, 1980
- Delayed sharing info structures, Witsenhausen 1971, Varaiya and Walrand, 1978, Mahajan, Nayyar, and Teneketzis 2010.
- Common past, Aicardi *et al* 1987
- Partially observed and partially nested, Casalino *et al* 1984
- Periodic sharing info structure, Ooi *et al* 1997
- Tower info structures, Swigart and Lall 2008
- Stochastic nested and belief sharing, Yüksel 2009
- P-classical and P-quasiclassical, Mahajan and Yüksel, 2010

Current state of affairs

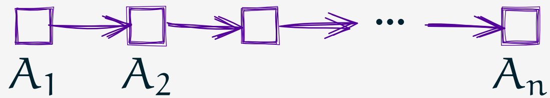
- Decentralized systems with non-classical information structures are studied on a **case-by-case basis**.
- Results are **hard to generalize** for even a slightly different setup

Develop a **systematic** methodology to derive structure of optimal decentralized control policies

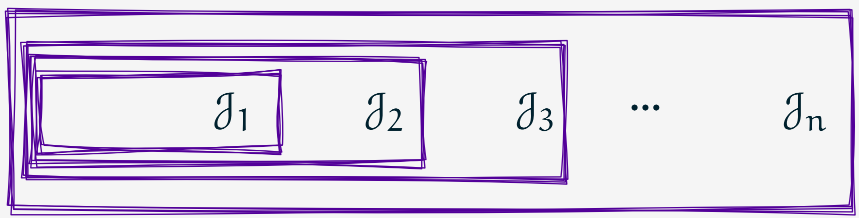
Overview of centralized stochastic control

Centralized stochastic control

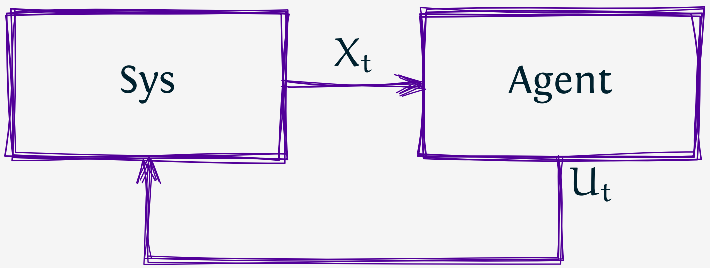
Single decision maker



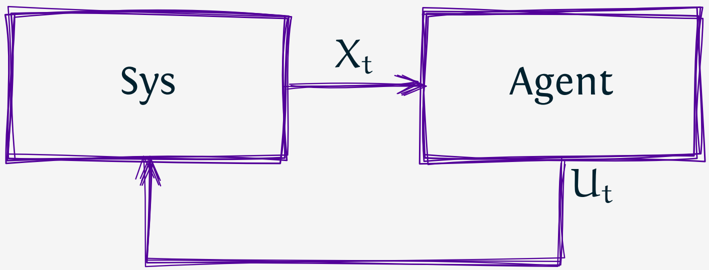
with classical information structures



MDP: Structural properties



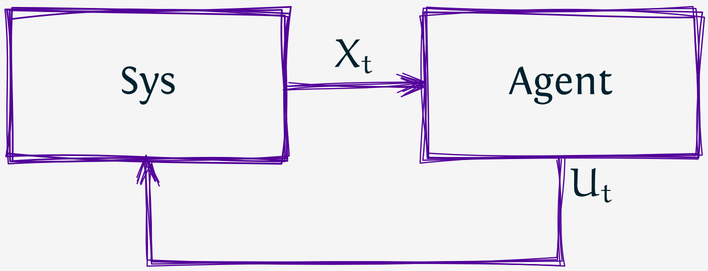
MDP: Structural properties

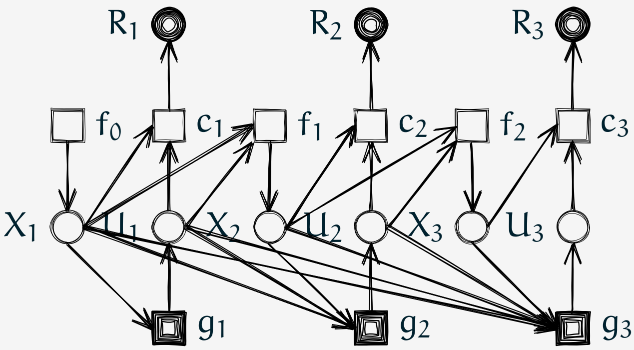


Structure of optimal policy

Choose current action based on **current state X_t**

MDP: Structural properties

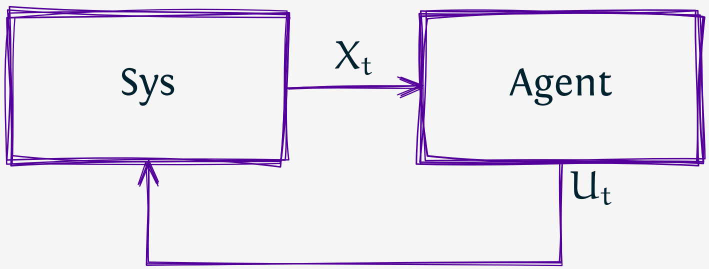


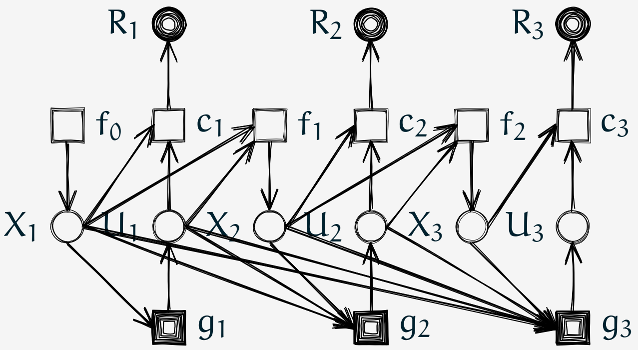


Structure of optimal policy

Choose current action based on **current state X_t**

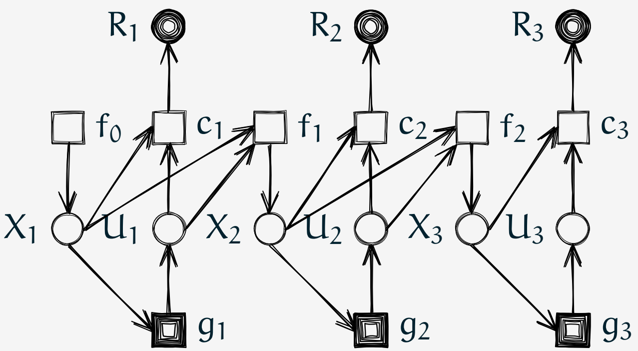
MDP: Structural properties



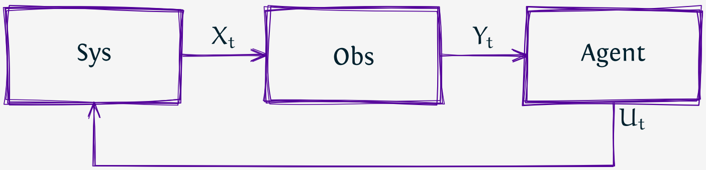


Structure of optimal policy

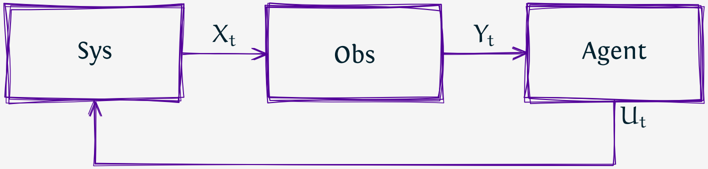
Choose current action based on **current state X_t**



POMDP: Structural properties



POMDP: Structural properties

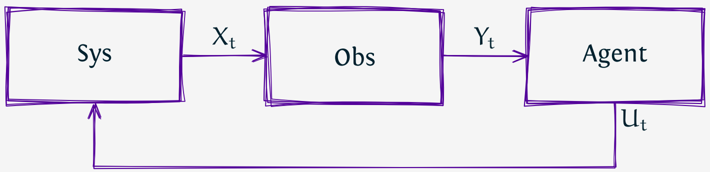


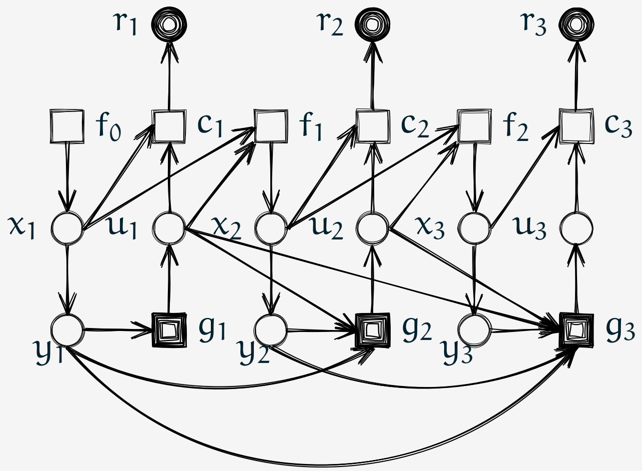
Structure of optimal policies

Choose current action
based on **current info state**

Pr(state of system | all data at agent)

POMDP: Structural properties



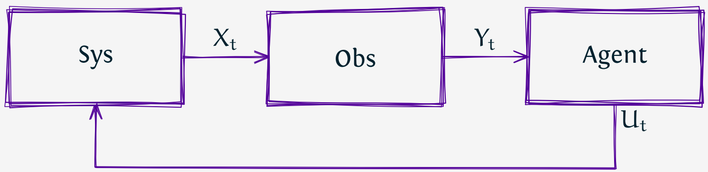


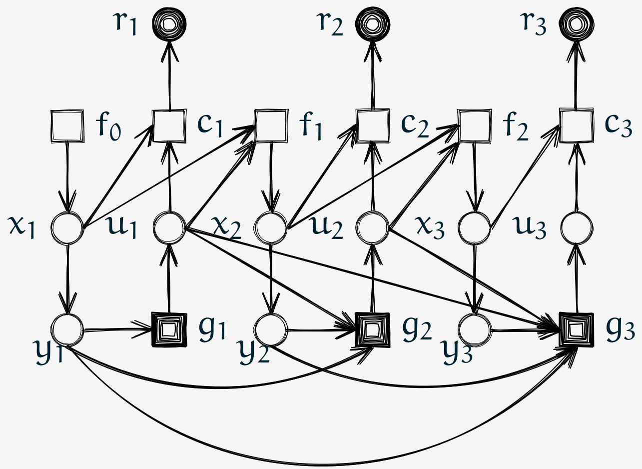
Structure of optimal policies

Choose current action
based on **current info state**

Pr(state of system | all data at agent)

POMDP: Structural properties

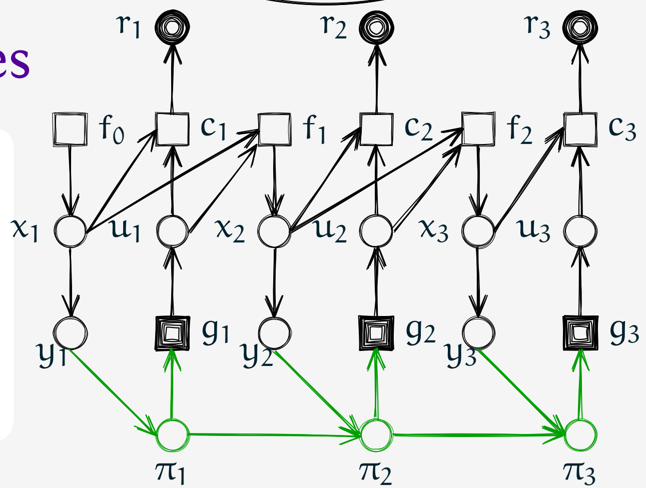




Structure of optimal policies

Choose current action
based on **current info state**

Pr(state of system | all data at agent)



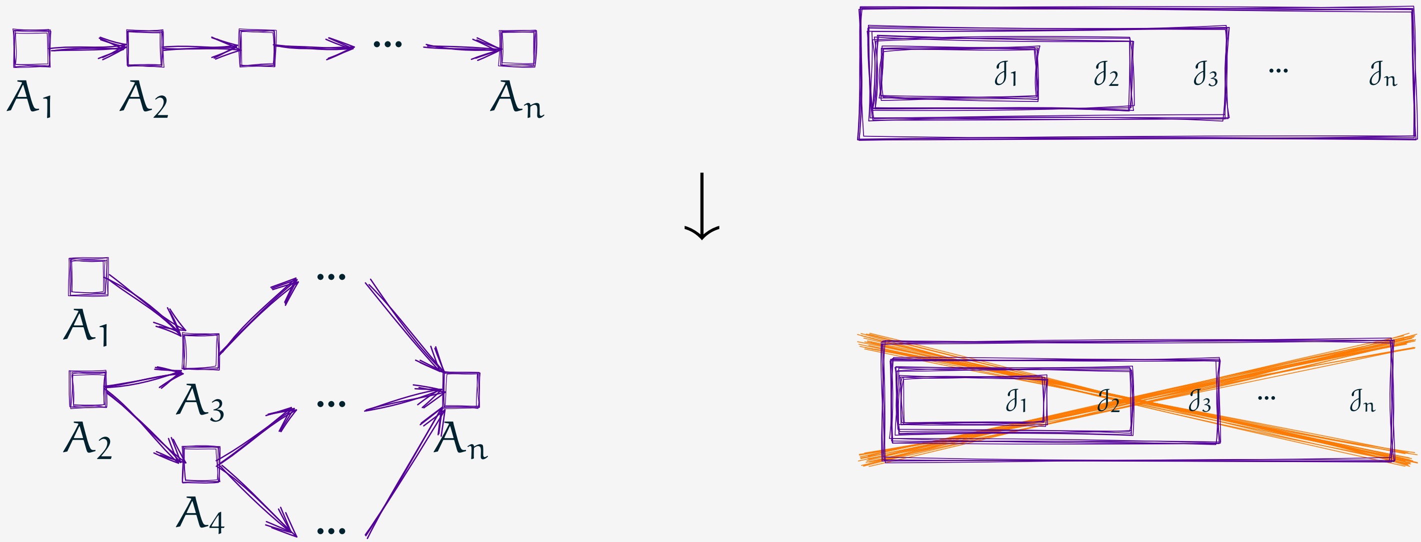
Structural policies in stochastic control

- Structure of optimal policies
  - **Shed irrelevant** information
  - **Compress relevant** information to a compact statistic
  - Hopefully, the data at the agent is not increasing with time

Structural policies in stochastic control

- Structure of optimal policies
  - **Shed irrelevant** information
  - **Compress relevant** information to a compact statistic
  - Hopefully, the data at the agent is not increasing with time
- Implication of the results
  - Simplify the functional form of the decision rules
  - Simplify search for optimal decision rules
  - A prerequisite for deriving dynamic programming decomposition.

Extending ideas to decentralized control

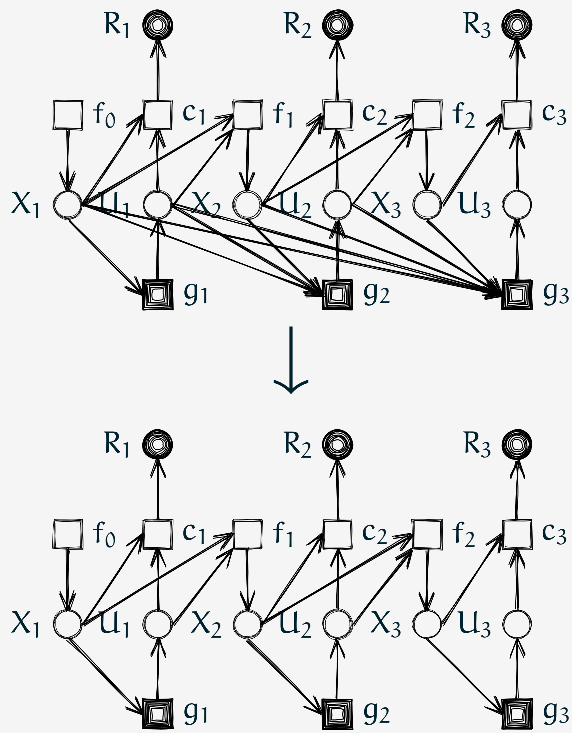


Outline

1. Why are decentralized systems difficult: an example
2. Overview of decentralized systems
   - Classification
   - Literature overview
3. Overview of Markov decision theory
4. **Systematic derivation of structural properties**
   - **Shed irrelevant information**
   - **Compress common information**
5. Automated derivation using graphical models
6. Conclusion

Shedding irrelevant information

Can we generalize the reasoning of MDPs to decentralized systems



The textbook proof

Define: $V_t(x_1, \dots, x_t) = \min\limits_{\text{all policies}} \mathbb{E}^g \left\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x^t \right\}$

Define: $W_t(x_t) = \min\limits_{\text{policies with req. structure}} \mathbb{E}^g \left\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x_t \right\}$

By definition: $W_t(x_t) \geqslant V_t(x_1, \dots, x_t)$ for any $x_1, \dots, x_t$.

Recursively prove: $W_t(x_t) \leqslant V_t(x_t, \dots, x_t)$ for any $x_1, \dots, x_t$.

The textbook proof

Define: $V_t(x_1, \ldots, x_t) = \min\limits_{\text{all policies}} \mathbb{E}^g \left\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x^t \right\}$

Define: $W_t(x_t) = \min\limits_{\text{policies with req. structure}} \mathbb{E}^g \left\{ \sum_{s=t}^{T} c(X_s, U_s) \mid x_t \right\}$

By definition: $W_t(x_t) \geqslant V_t(x_1, \ldots, x_t)$ for any $x_1, \ldots, x_t$.

Recursively prove: $W_t(x_t) \leqslant V_t(x_t, \ldots, x_t)$ for any $x_1, \ldots, x_t$.

$$W_t(x_t) = V_t(x_1, \ldots, x_t) \text{ for all } x_1, \ldots, x_t$$

The textbook proof

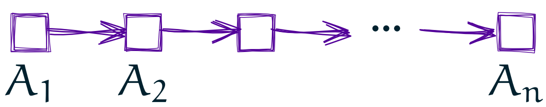
De

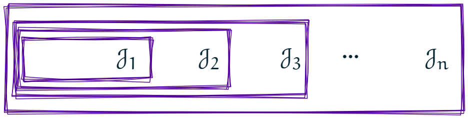
De

By

Re

Proof tied to centralized system . . .





$$W_t(x_t) = V_t(x_1, ..., x_t) \text{ for all } x_1, ..., x_t$$

Is there a proof that can be extended to decentralized systems?

The main idea

Suppose we have to minimize cost from the p.o.v. of one agent and

$$\mathrm{E}[\text{cost} \mid \text{ all data }] = \mathrm{F}(\text{relevant data}, \text{control action})$$

The main idea

Suppose we have to minimize cost from the p.o.v. of one agent and

$$\mathbb{E}[\text{cost} \mid \text{ all data }] = F(\text{relevant data}, \text{control action})$$

Without loss of optimality, choose
control action $= g(\text{relevant data})$.

The main idea

Suppose we have to minimize cost from the p.o.v. of one agent and

$$\mathrm{E}[\text{cost} \mid \text{ all data }] = \mathrm{F}(\text{relevant data}, \text{control action})$$

Without loss of optimality, choose control action $=$ g(relevant data).

Rest is just a matter of detail.

The main idea

Step 1. Pick an agent

Step 2. If the agent observes any **irrelevant data**, ignore those observations

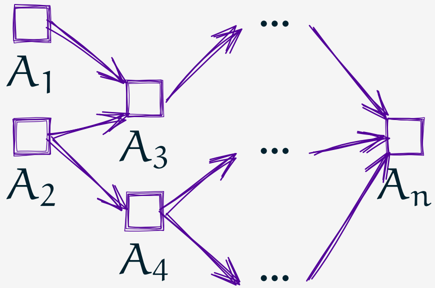
Step 3. Repeat

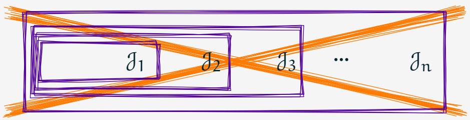
The main idea

Step 1. Pick an agent

Step 2. If the agent observes any **irrelevant data**, ignore those observations

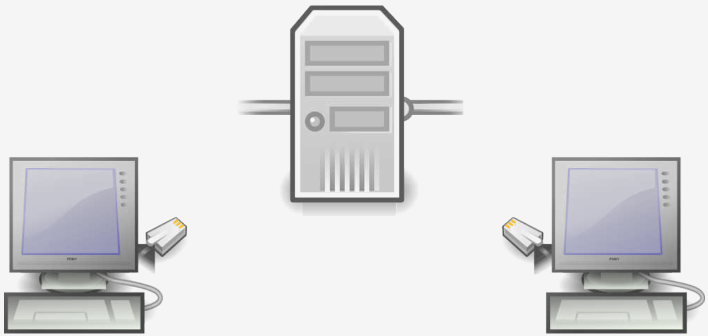
Step 3. Repeat





Is easy to extend to decentralized systems . . .

Multiaccess broadcast



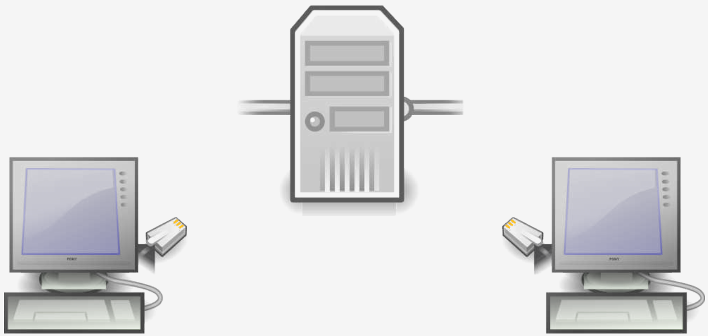
$$U_t^i = g_{i,t}(X^i_{1:t}, U^i_{1:t-1}, Z_{1:t-1})$$

X_t = state of queue

U_t = Tx or not

Z_t = Channel feedback

Multiaccess broadcast



$$U_t^i = g_{i,t}(X_{1:t}^i, U_{1:t-1}^i, Z_{1:t-1})$$

X_t = state of queue

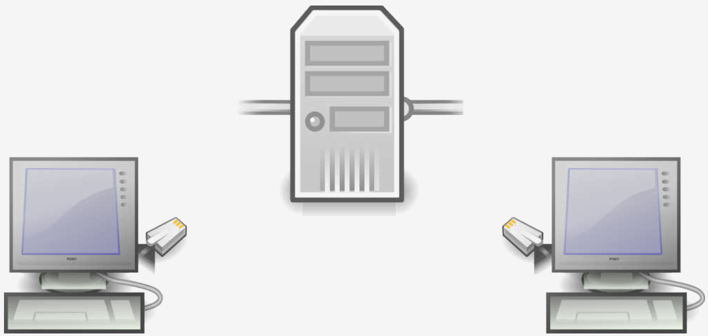
U_t = Tx or not

Z_t = Channel feedback

Remove irrelevant data

Conditioned on $(X_t^i, U_t^i, Z_{1:t-1})$, the future reward $R_{t+1:T}$ is independent of past $(X_{1:t-1}^i, U_{1:t-1}^i)$.

Multiaccess broadcast



$$U_t^i = g_{i,t}(X_{1:t}^i, U_{1:t-1}^i, Z_{1:t-1})$$

X_t = state of queue

U_t = Tx or not

Z_t = Channel feedback

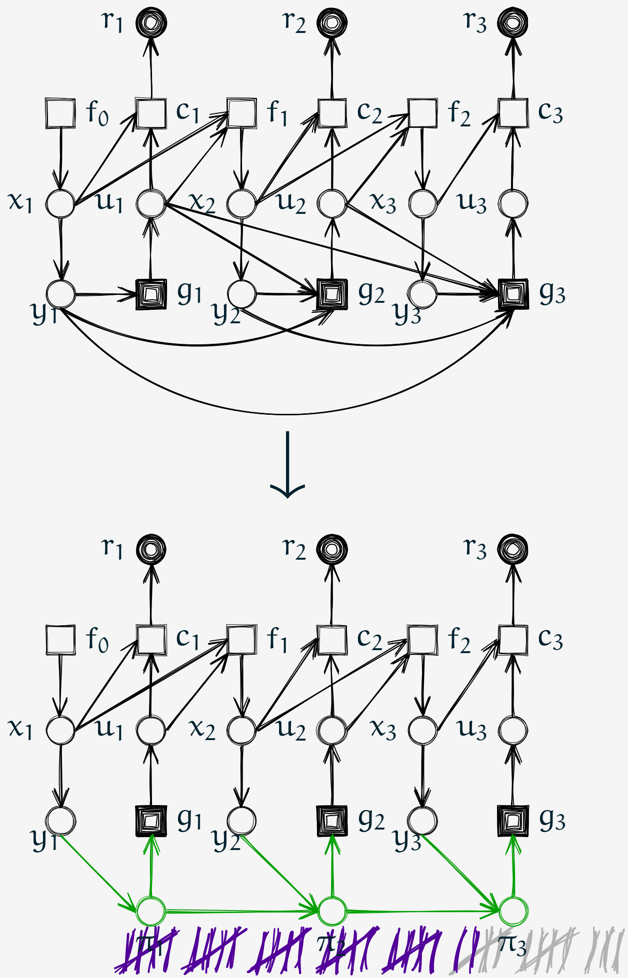
Remove irrelevant data

Conditioned on $(X_t^i, U_t^i, Z_{1:t-1})$, the future reward $R_{t+1:T}$ is independent of past $(X_{1:t-1}^i, U_{1:t-1}^i)$.

$$U_t^i = g_{i,t}(X_t^i, Z_{1:t-1})$$

Compressing relevant information

Can we generalize the reasoning of POMDPs to decentralized systems



The textbook proof

- Find sufficient statistic for performance analysis

$$\pi_t = \Pr(\text{ state } \mid \text{ all data })$$

The textbook proof

- Find sufficient statistic for performance analysis

$$\pi_t = \Pr(\text{ state } \mid \text{ all data })$$

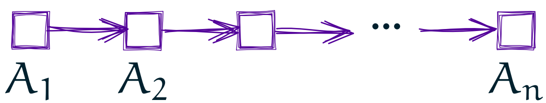
- This sufficient statistic can be updated recursively!

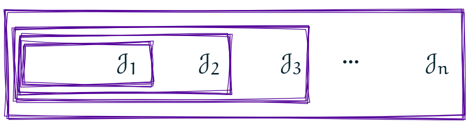
$$\pi_{t+1} = F(\pi_t, U_t, Y_t)$$

The textbook proof

- Find sufficient statistic for performance analysis

Proof tied to centralized system . . .





Solution Approach

- Given a group of agents, their **coordinator** observes data that is commonly available at all agents and tells each agent what to do with its private data.

Solution Approach

- Given a group of agents, their **coordinator** observes data that is commonly available at all agents and tells each agent what to do with its private data.
- Optimal design of the coordinator is **equivalent** to the optimal design of all agents in the group.

Solution Approach

- Given a group of agents, their **coordinator** observes data that is commonly available at all agents and tells each agent what to do with its private data.
- Optimal design of the coordinator is **equivalent** to the optimal design of all agents in the group.
- If the data at each agent in the group is increasing with time, **the problem at the coordinator is centralized**.

Solution Approach

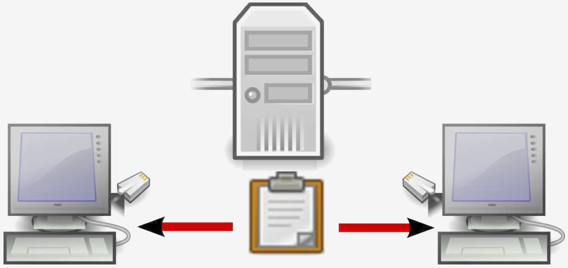
- Given a group of agents, their **coordinator** observes data that is commonly available at all agents and tells each agent what to do with its private data.
- Optimal design of the coordinator is **equivalent** to the optimal design of all agents in the group.
- If the data at each agent in the group is increasing with time, **the problem at the coordinator is centralized**.
- Use results from POMDP to compress the data at the controller to a sufficient statistic. That is also **a sufficient statistic for the commonly observed data** for each agent in the group.

Multiaccess broadcast

$$U_t^1 = g_{1,t}(X_t^1, Z_{1:t-1})$$

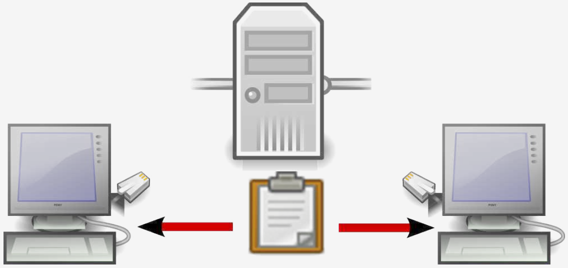
$$U_t^2 = g_{2,t}(X_t^2, Z_{1:t-1})$$

Coordinator of a system



$$u_t^1 = g_{1,t}(X_t^1, Z_{1:t-1})$$
$$u_t^2 = g_{2,t}(X_t^2, Z_{1:t-1})$$

Coordinator of a system

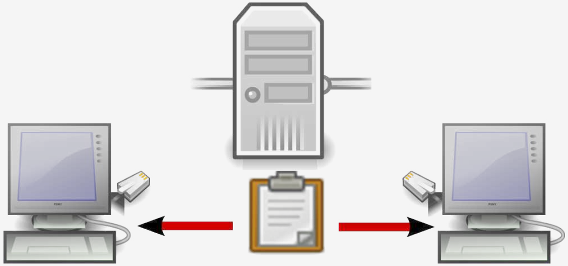


$$U_t^1 = g_{1,t}(X_t^1, \boxed{Z_{1:t-1}})$$
$$U_t^2 = g_{2,t}(X_t^2, \boxed{Z_{1:t-1}})$$

Chooses partial functions

$$(\gamma_t^1, \gamma_t^2) = \psi(Z_{1:t-1}), \qquad \gamma_t^i : \mathcal{X}^i \to \mathcal{U}$$

Coordinator of a system



$$U_t^1 = g_{1,t}(X_t^1, \boxed{Z_{1:t-1}})$$
$$U_t^2 = g_{2,t}(X_t^2, \boxed{Z_{1:t-1}})$$

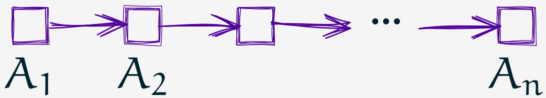
Chooses partial functions

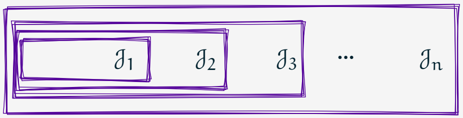
$$(\gamma_t^1, \gamma_t^2) = \psi(Z_{1:t-1}), \qquad \gamma_t^i : \mathcal{X}^i \to \mathcal{U}$$

The agents simply use the partial function

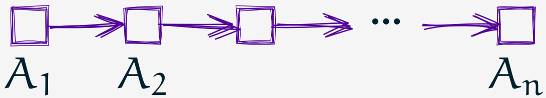
$$U_t^1 = \gamma_t^1(X_t^1) \qquad U_t^2 = \gamma_t^2(X_t^2)$$

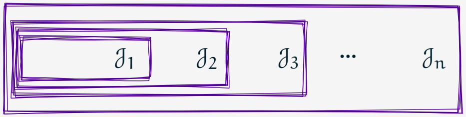
The coordinator's problem is centralized





The coordinator's problem is centralized

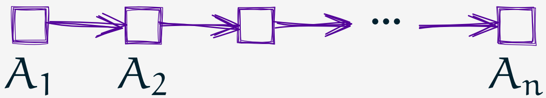


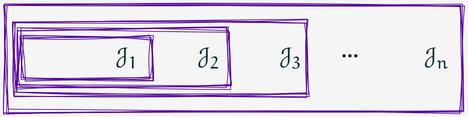


Structure of coordinator's policy

$$(\gamma^1_t, \gamma^2) = \psi_t(\pi_t), \qquad \pi_t = \Pr(X^1_t, X^2_t \mid Z_{1:t-1})$$

The coordinator's problem is centralized





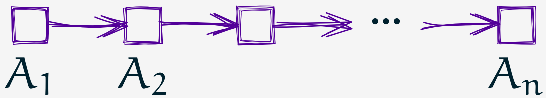
Structure of coordinator's policy

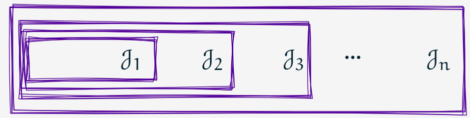
$$(\gamma_t^1, \gamma^2) = \psi_t(\pi_t), \qquad \pi_t = \Pr(X_t^1, X_t^2 \mid Z_{1:t-1})$$

Structure of transmitter's policy

$$U_t^i = g_{i,t}(X_t^i, \pi_t)$$

The coordinator's problem is centralized





Structure of coordinator's policy

$$(\gamma_t^1, \gamma^2) = \psi_t(\pi_t), \qquad \pi_t = \Pr(X_t^1, X_t^2 \mid Z_{1:t-1})$$

Structure of transmitter's policy

$$U_t^i = g_{i,t}(X_t^i, \pi_t)$$

Can be used to obtain a **dynamic programming** decomposition.

Optimal solution

- For symmetric arrival rates p
  - If $p > \tau$, follow TDMA
  - If $p < \tau$,
    - S1. If you have a packet, transmit it. If collision, one user moves to S2.
    - S2. Idle, then move to S1

Optimal solution

- For symmetric arrival rates p
  - If $p > \tau$, follow TDMA
  - If $p < \tau$,
    - S1. If you have a packet, transmit it. If collision, one user moves to S2.
    - S2. Idle, then move to S1
- Same as the strategy proposed by Hluchyj and Gallager.

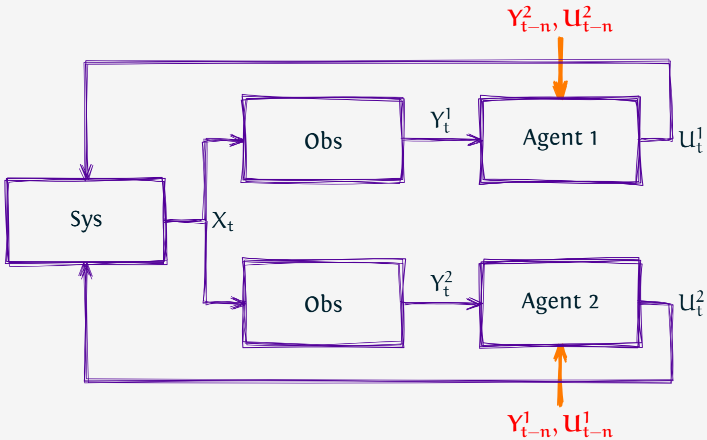
Optimal solution

- For symmetric arrival rates p
  - If $p > \tau$, follow TDMA
  - If $p < \tau$,
    - S1. If you have a packet, transmit it. If collision, one user moves to S2.
    - S2. Idle, then move to S1
- Same as the strategy proposed by Hluchyj and Gallager.

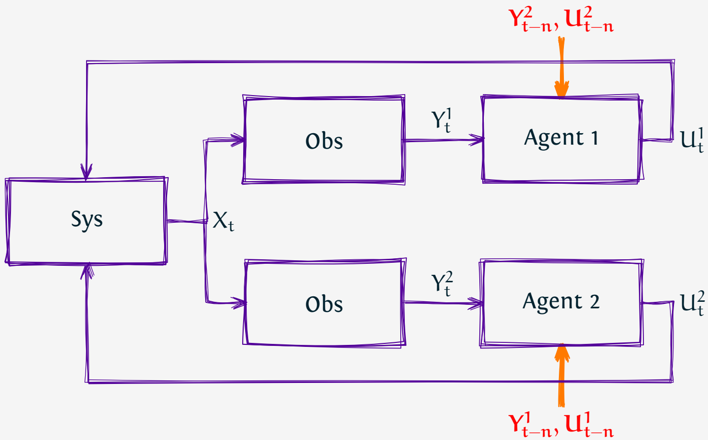
We can prove optimality. All previous attempts provide approximate solutions!

Application to other problems

Delayed sharing info structure (DSIS)

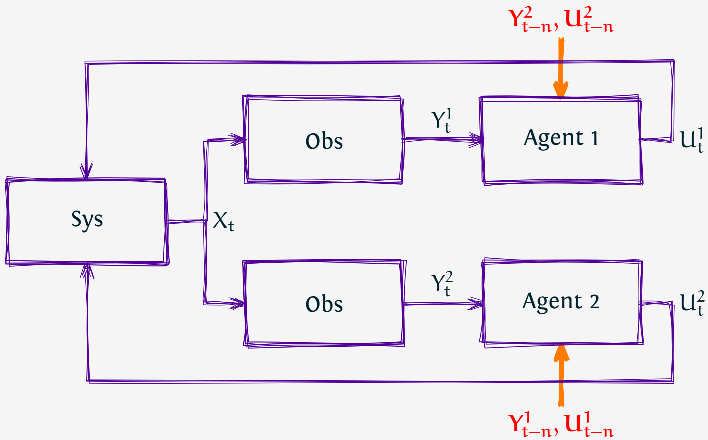


Delayed sharing info structure (DSIS)



- K controllers that **share information with a delay** of n time steps
- $n = 0 \Rightarrow$ classical info structure (centralized system)
- $n = \infty \Rightarrow$ non-classical info structure with no sharing (completely decentralized system)

Delayed sharing info structure (DSIS)

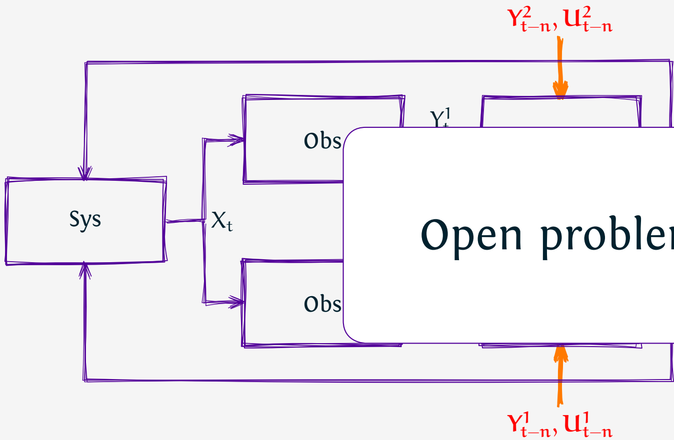


- K controllers that **share information with a delay** of n time steps
- $n = 0 \Rightarrow$ classical info structure (centralized system)
- $n = \infty \Rightarrow$ non-classical info structure with no sharing (completely decentralized system)

History

- **Witsenhausen, 1971** proposed the n-DSIS and **asserted** a structure of optimal control policies
- **Varaiya and Walrand, 1979** proved that Witsenhausen's assertion is **true for $n = 1$** but **false of $n > 1$**

Delayed sharing info structure (DSIS)



Open problem for 39 years

hat **share information**
of n time steps

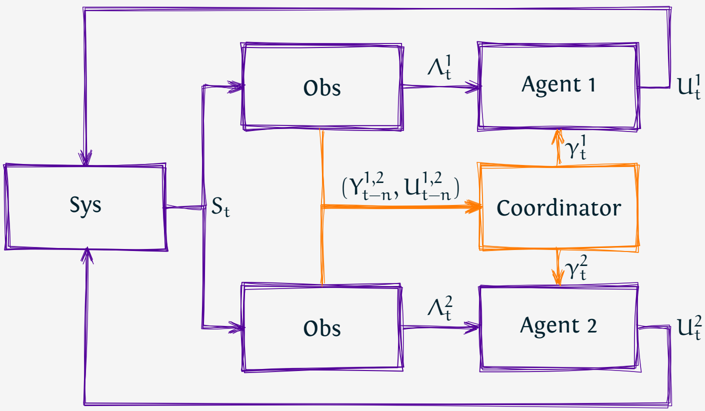
ssical info structure (centralized system)

- $n = \infty \Rightarrow$ non-classical info structure with no sharing (completely decentralized system)

History

- **Witsenhausen, 1971** proposed the n-DSIS and **asserted** a structure of optimal control policies
- **Varaiya and Walrand, 1979** proved that Witsenhausen's assertion is **true for $n = 1$** but **false of $n > 1$**

Delayed sharing info structure (DSIS)

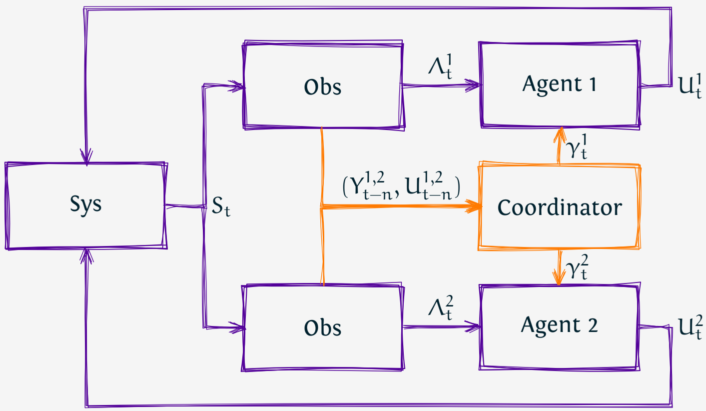


$$(\gamma_t^1, \gamma_t^2) = \psi_t(\text{common info})$$

$$U_t^1 = \gamma_t^1(\text{private info})$$

$$U_t^2 = \gamma_t^2(\text{private info})$$

Delayed sharing info structure (DSIS)



$$(\gamma_t^1, \gamma_t^2) = \psi_t(\text{common info})$$

$$U_t^1 = \gamma_t^1(\text{private info})$$

$$U_t^2 = \gamma_t^2(\text{private info})$$

Structural properties

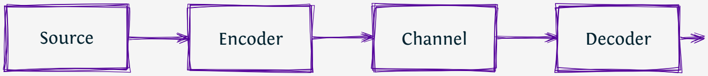
The coordinator's problem is centralized.

Can derive structure of optimal control policies.

Nayyar-M-Teneketzis, TAC 10

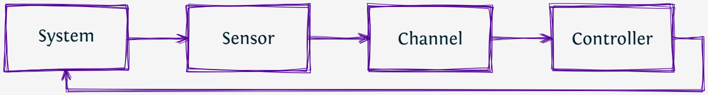
Other examples

Real-time communication



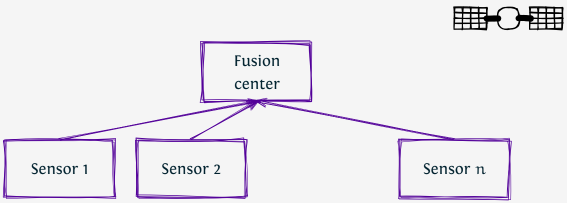
M-Teneketzis, TIT 09

Control over noisy channels



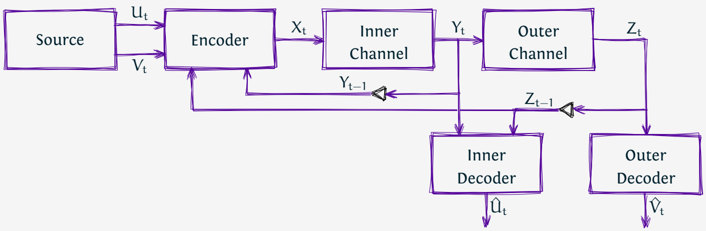
M-Teneketzis, SICON 09

Sensor scheduling



Shuman-Nayyar-M-*et al.* Proc IEEE, 10, JSTARS 10

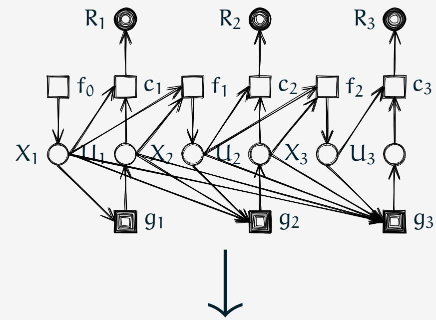
Broadcast with feedback

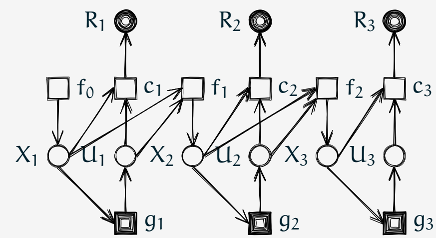


M, Allerton 09

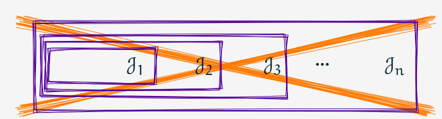
Summary of proposed method

Shedding irrelevant information



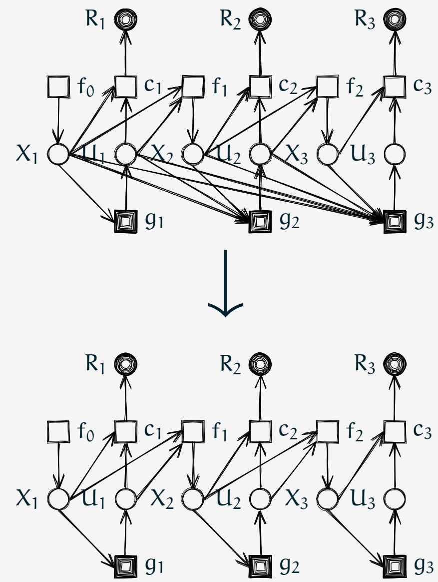


applied to

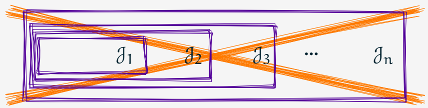


Shedding irrelevant information

- Iterative procedure
  - Shed irrelevant data at **an agent** (at a particular time)
  - Iterate over all agents until a fixed point

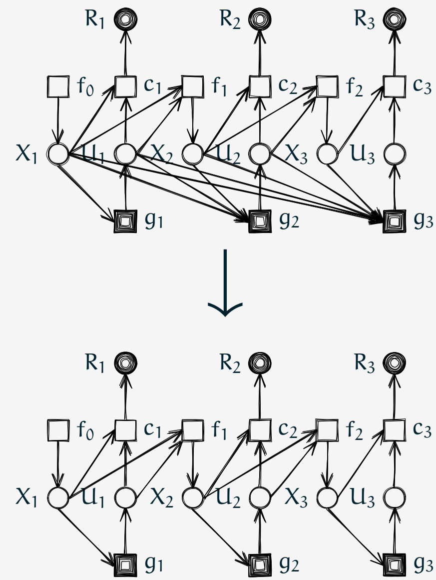


applied to

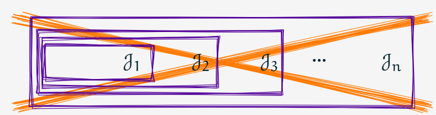


Shedding irrelevant information

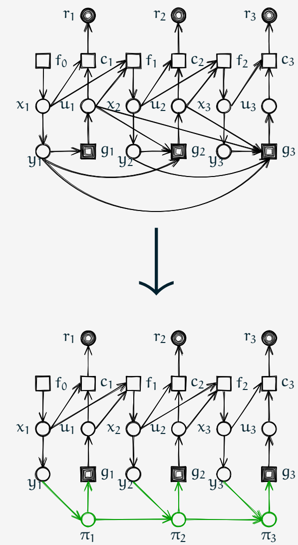
- Iterative procedure
  - Shed irrelevant data at **an agent** (at a particular time)
  - Iterate over all agents until a fixed point
  - Repeat for all **coordinators** of groups of agents



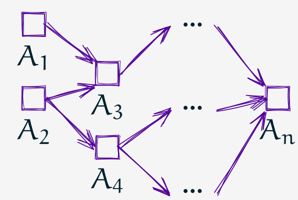
applied to



Compressing relevant information

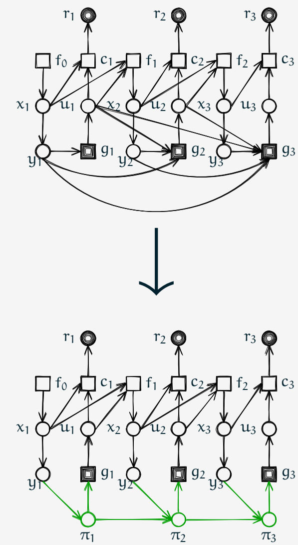


applied to

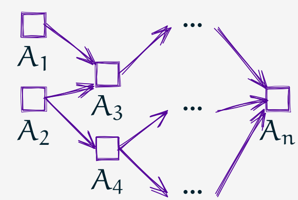


Compressing relevant information

- Iterative procedure
  - Find **common information** between a group of agents
  - Look at the problem from the p.o.v. of a **coordinator** that observes this common info, and chooses **partial functions**.

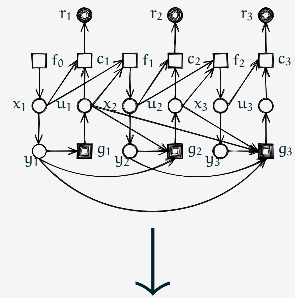


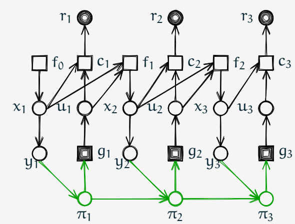
applied to



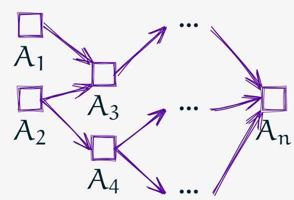
Compressing relevant information

- Iterative procedure
  - Find **common information** between a group of agents
  - Look at the problem from the p.o.v. of a **coordinator** that observes this common info, and chooses **partial functions**.
  - Repeat for all **groups of agents**





applied to



Automating the procedure

Automating the procedure

Automating the procedure

- Irrelevant data, dependent rewards, conditional independence

Automating the procedure

- Irrelevant data, dependent rewards, conditional independence

Directed acyclic graphs and graphical models

Automating the procedure

- Irrelevant data, dependent rewards, conditional independence

Directed acyclic graphs and graphical models

- Common information, state for input-output mapping

Automating the procedure

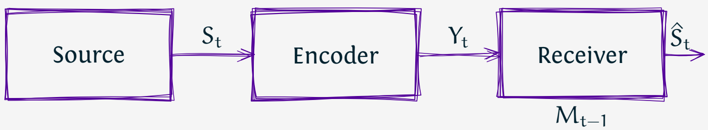
- Irrelevant data, dependent rewards, conditional independence

Directed acyclic graphs and graphical models

- Common information, state for input-output mapping

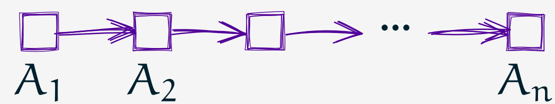
Information lattice and cuts of a lattice

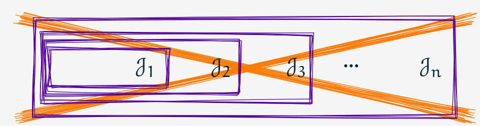
An example: Real-time communication



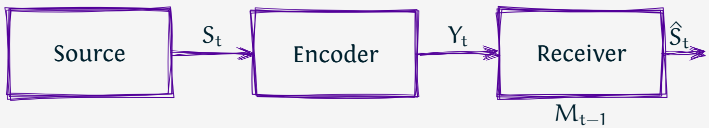
- Hans S. Witsenhausen, **On the structure of real-time source coders**, BSJT-79.

Markov source	$S_{t+1} = f_t(S_t, W_t)$
Causal encoder	$Y_t = c_t(S_{1:t}, Y_{1:t-1})$
Finite memory decoder	$\hat{S}_t = g_t(Y_t, M_{t-1})$
memory update	$M_t = l_t(Y_t, M_{t-1})$
Minimize distortion	$\min \mathbb{E}\left\{ \sum_{t=1}^{T} \rho_t(S_t, \hat{S}_t) \right\}$





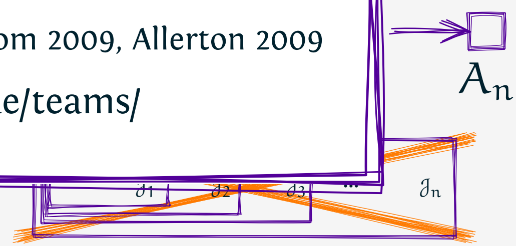
An example: Real-time communication



- Hans S. Witsenhausen, **On the structure of real-time source coders**, BSJT-79.

Automatic derivation of structural results using graphical models — M-Tatikonda, ConCom 2009, Allerton 2009

http://pantheon.yale.edu/ am894/code/teams/



Minimize distortion $\min \mathbb{E}\left\{ \sum_{t=1} \rho_t(S_t, \hat{S}_t) \right\}$

Description of the problem

```
s  = mkNonReward "s"    ; ŝ  = mkNonReward "ŝ"
y  = mkNonReward "y"    ; m  = mkNonReward "m"
r  = mkReward    "r"

f = mkStochastic "f"    ; c = mkControl    "c"
g = mkControl    "g"    ; l = mkControl    "l"
d = mkStochastic "d"

dynamics t | t == 1 = f(1).$.(s(1) .|. [])
                   ++ c(1).$.(y(1) .|. [s(1)])
                   ++ g(1).$.(ŝ(1) .|. [y(1)])
                   ++ l(1).$.(m(1) .|. [y(1)])
                   ++ d(1).$.(r(1) .|. [s(1), ŝ(1)])
           | otherwise = f(t).$.(s(t) .|. [s(t-1)])
                      ++ c(t).$.(y(t) .|. map s[1..t] ++ map y[1..t-1])
                      ++ g(t).$.(ŝ(t) .|. [y(t), m(t-1)])
                      ++ l(t).$.(m(t) .|. [y(t), m(t-1)])
                      ++ d(t).$.(r(t) .|. [s(t), ŝ(t)])

rt = mkTeamTime dynamics 3
```

Verifying the model

```
*Data.Teams.Examples.Wit79> printTeam rt
Stochastic:
===========
f1.$.([s1].|.[])
d1.$.([r1].|.[s1, ŝ1])
f2.$.([s2].|.[s1])
d2.$.([r2].|.[s2, ŝ2])
f3.$.([s3].|.[s2])
d3.$.([r3].|.[ŝ3, s3])


Control :
=========
y1 = c1([s1])
ŝ1 = g1([y1])
m1 = l1([y1])
y2 = c2([y1, s2, s1])
ŝ2 = g2([y2, m1])
m2 = l2([y2, m1])
y3 = c3([s3, y2, y1, s2, s1])
ŝ3 = g3([m2, y3])
m3 = l3([m2, y3])
```

Simplifying the model

```
*Data.Teams.Examples.Wit79> printTeam (simplify rt)
Stochastic:
===========
f1.$.([s1].|.[])
d1.$.([r1].|.[s1, ŝ1])
f2.$.([s2].|.[s1])
d2.$.([r2].|.[s2, ŝ2])
f3.$.([s3].|.[s2])
d3.$.([r3].|.[ŝ3, s3])

Control :
=========
y1 = c1([s1])
ŝ1 = g1([y1])
m1 = l1([y1])
y2 = c2([m1, s2])
ŝ2 = g2([m1, y2])
m2 = l2([m1, y2])
y3 = c3([m2, s3])
ŝ3 = g3([m2, y3])
m3 = l3([])
```

Systematic design of decentralized systems

Structure of optimal policies

The data at the controllers increases with time, leading to a doubly exponential increase in the number of policies.

When can an agent, or a group of agents,

- shed available information
- compress available information

without loss of optimality?

Search of optimal policies

- Brute force search of an optimal policy has doubly exponential complexity with time-horizon.
- How can we search for an optimal policy efficiently?
- How can we implement an optimal policy efficiently?

Design principles

- Can we check if the optimal design of a decentralized system is tractable, **without actually designing the system**?
- Can we provide additional information to agents to make the design tractable? If so, can we find the smallest such information?

Reflections

- Non-sequential information structures
  - **Conceptual difficulties**
  - Computational difficulties

Reflections

- Non-sequential information structures
  - **Conceptual difficulties**
  - Computational difficulties
- Provides high-level design guidelines
  - The optimal solution needs to computed **numerically**
  - Provides some design insights: structural properties, which modeling assumption makes the problem easier, etc.

Reflections

- Non-sequential information structures
  - **Conceptual difficulties**
  - Computational difficulties
- Provides high-level design guidelines
  - The optimal solution needs to computed **numerically**
  - Provides some design insights: structural properties, which modeling assumption makes the problem easier, etc.
- Actual solution requires simplification and approximation based on “domain knowledge”

Thank you