Simplification of sequential teams

Aditya Mahajan
Yale University

Joint work with Sekhar Tatikonda

Acknowledements: Demos Teneketzis
and Ashutosh Nayyar (Univ of Michigan)


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

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.
- However, most 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.
- However, most 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.
- However, most 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?

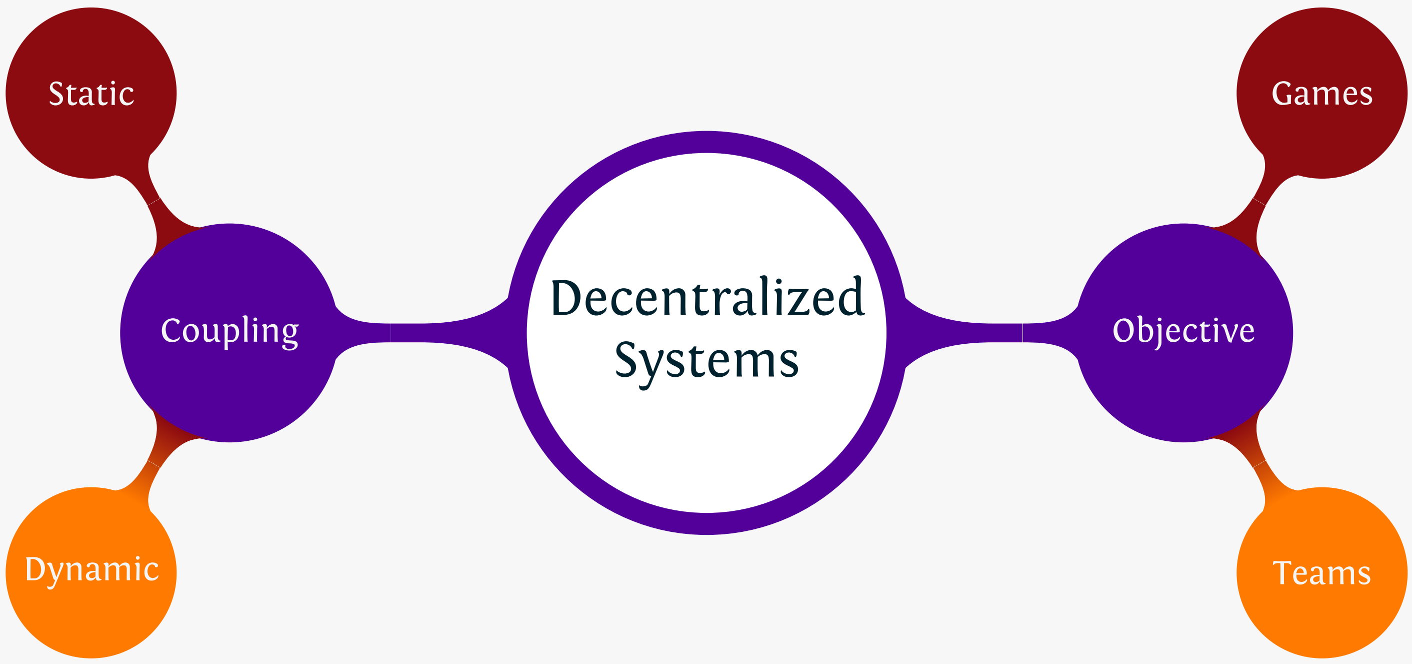
Design principles

- 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?

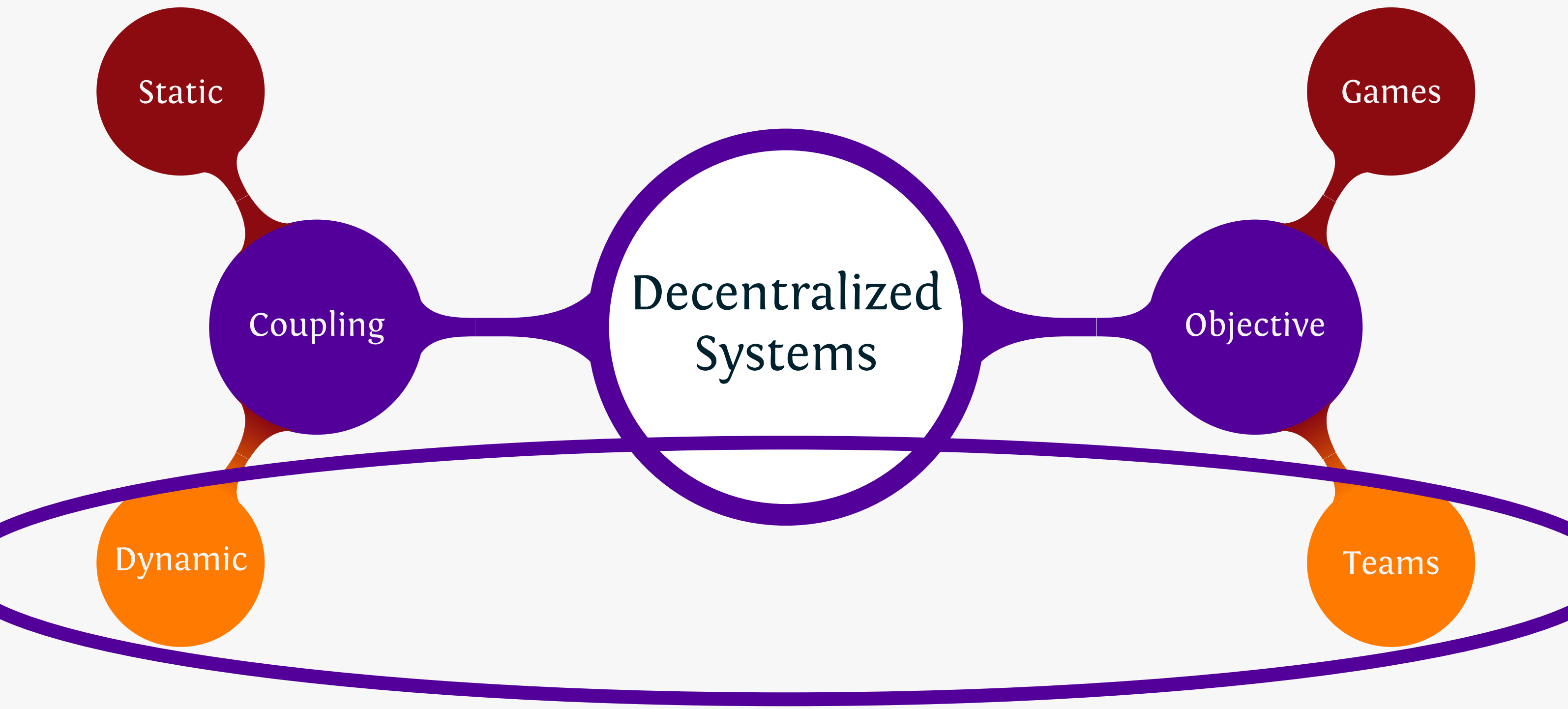
Outline

1. Overview of decentralized systems
2. Systematic derivation of structural properties
   - Shed irrelevant information
   - Compress common information
3. Automated derivation using graphical models

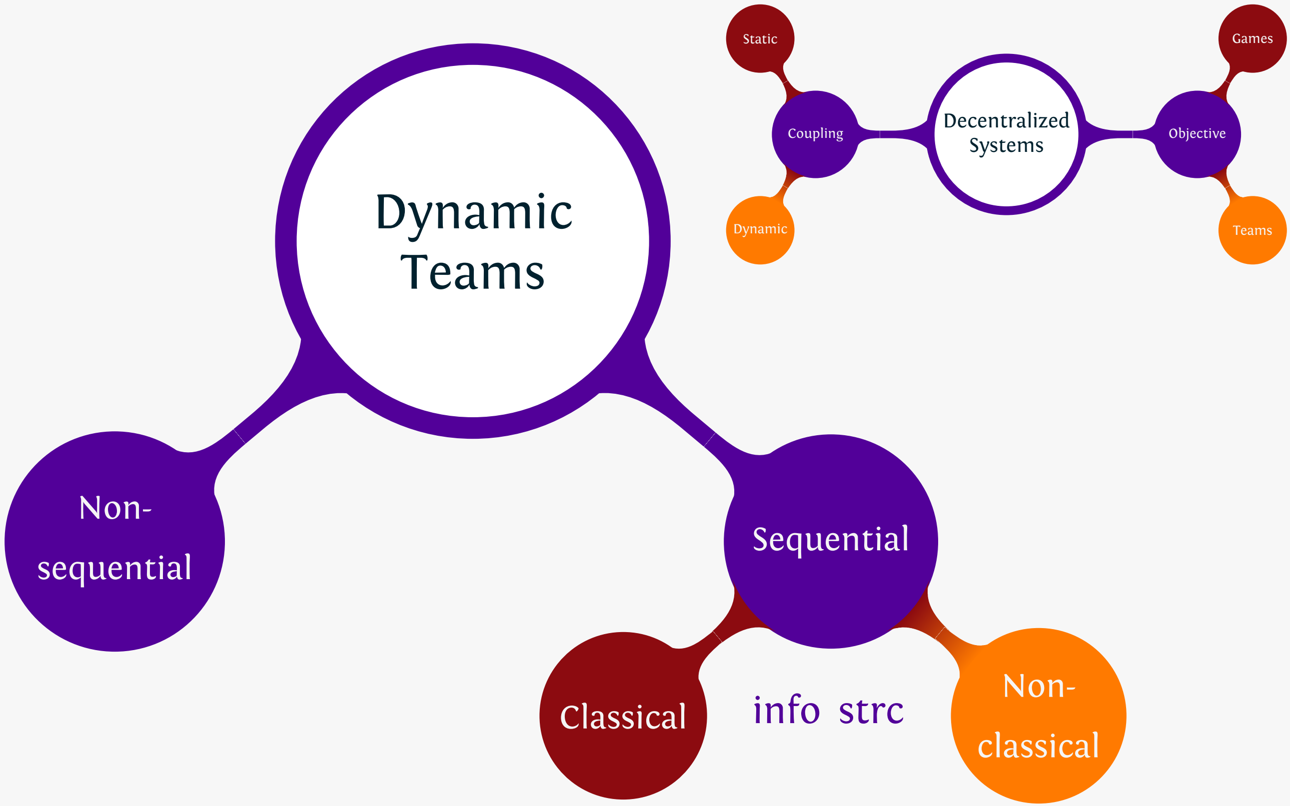
Classification of decentralized systems



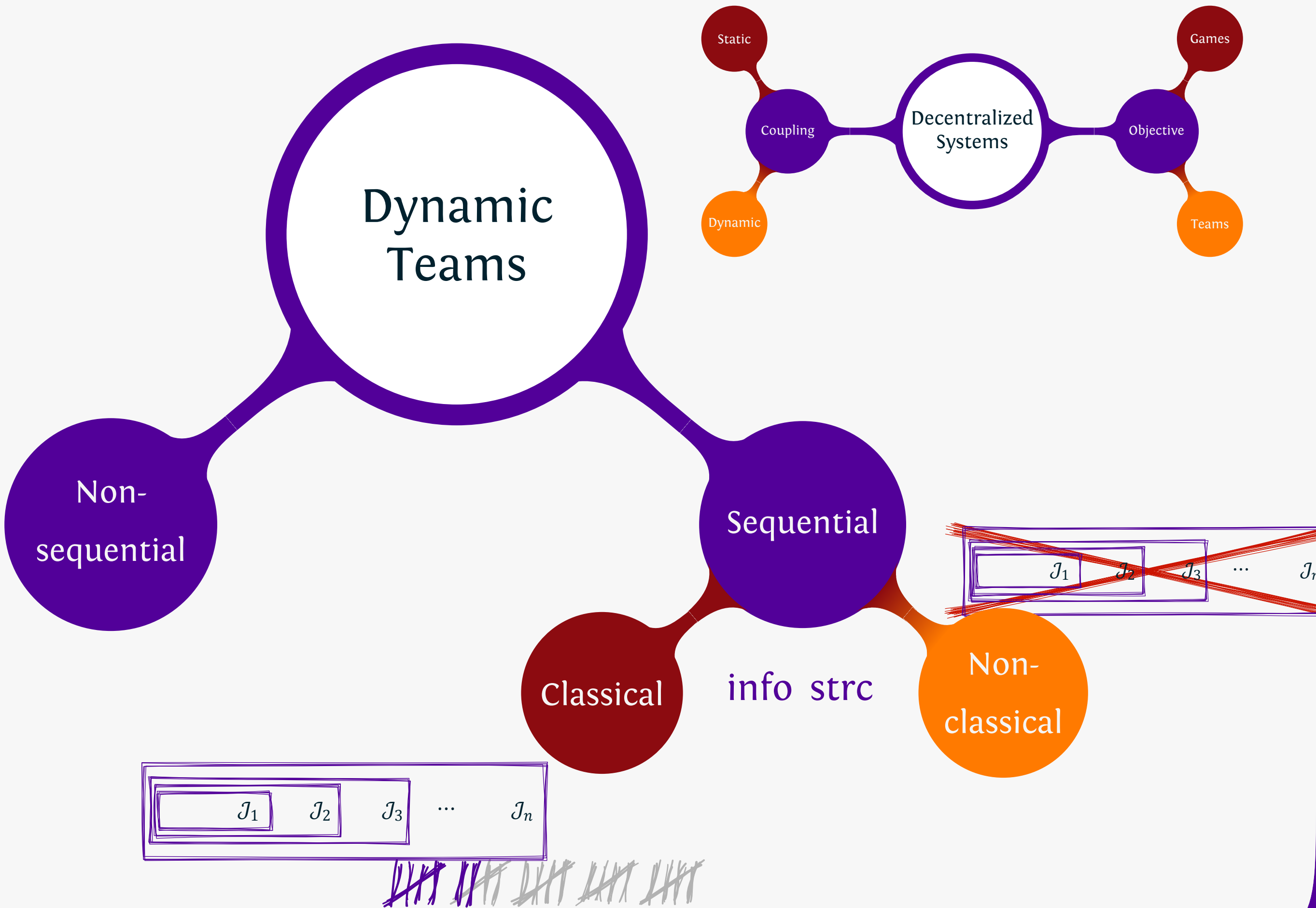
Classification of decentralized systems



Classification of decentralized systems

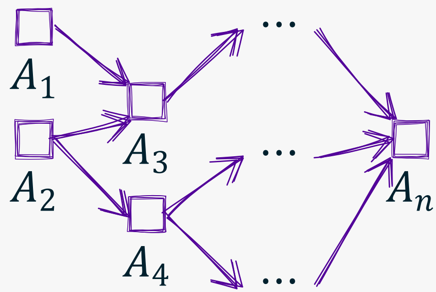


Classification of decentralized systems

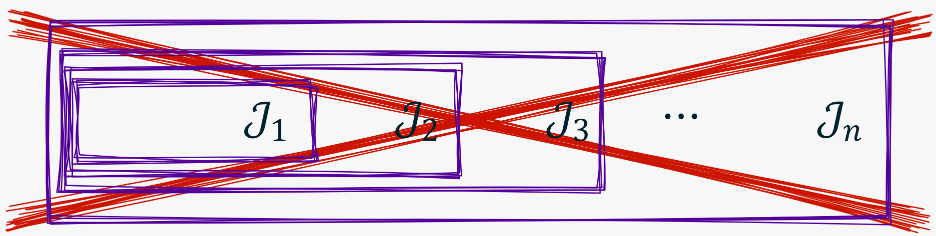


We are interested in

Sequential dynamic teams



with non-classical information structures



Current state of affairs

- Been an active area of research for almost 50 years . . .
- 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

System model

- System Variables $(X_1, \ldots, X_n)$.

System model

- System Variables $(X_1, \dots, X_n)$.
  - Control variables $A \subset N = \{1, \dots, n\}$. A decision maker chooses X_α, $\alpha \in A$.
  - stochastic variables $M = N \setminus A$. Nature chooses X_m, $m \in M$.

System model

- System Variables $(X_1, \ldots, X_n)$.
  - Control variables $A \subset N = \{1, \ldots, n\}$. A decision maker chooses X_α, $\alpha \in A$.
  - stochastic variables $M = N \setminus A$. Nature chooses X_m, $m \in M$.
- Measurable spaces $(\mathcal{X}_k, \mathfrak{F}_k)$, $k \in N$.

System model

- System Variables $(X_1, \ldots, X_n)$.
  - Control variables $A \subset N = \{1, \ldots, n\}$. A decision maker chooses X_α, $\alpha \in A$.
  - stochastic variables $M = N \setminus A$. Nature chooses X_m, $m \in M$.
- Measurable spaces $(\mathcal{X}_k, \mathfrak{F}_k)$, $k \in N$.
- Information sets $I_k \subset \{1, \ldots, k-1\}$, $k \in N$.

System model

- System Variables $(X_1, \ldots, X_n)$.
  - Control variables $A \subset N = \{1, \ldots, n\}$. A decision maker chooses X_α, $\alpha \in A$.
  - stochastic variables $M = N \setminus A$. Nature chooses X_m, $m \in M$.
- Measurable spaces $(\mathcal{X}_k, \mathfrak{F}_k)$, $k \in N$.
- Information sets $I_k \subset \{1, \ldots, k-1\}$, $k \in N$.
  - Control law $g_\alpha : (\mathcal{X}_{I_\alpha}, \mathfrak{F}_{I_\alpha}) \mapsto (\mathcal{X}_\alpha, \mathfrak{F}_\alpha)$, $\alpha \in A$.
  - Stochastic kernel p_m from $(\mathcal{X}_{I_m}, \mathfrak{F}_{I_m})$ to $(\mathcal{X}_m, \mathfrak{F}_m)$.

System model

- System Variables $(X_1, \dots, X_n)$.
  - Control variables $A \subset N = \{1, \dots, n\}$. A decision maker chooses X_α, $\alpha \in A$.
  - stochastic variables $M = N \setminus A$. Nature chooses X_m, $m \in M$.
- Measurable spaces $(\mathcal{X}_k, \mathfrak{F}_k)$, $k \in N$.
- Information sets $I_k \subset \{1, \dots, k-1\}$, $k \in N$.
  - Control law $g_\alpha : (\mathcal{X}_{I_\alpha}, \mathfrak{F}_{I_\alpha}) \mapsto (\mathcal{X}_\alpha, \mathfrak{F}_\alpha)$, $\alpha \in A$.
  - Stochastic kernel p_m from $(\mathcal{X}_{I_m}, \mathfrak{F}_{I_m})$ to $(\mathcal{X}_m, \mathfrak{F}_m)$.
- Reward variables $R \subset N$.

$$\max_{g_\alpha, \alpha \in A} \mathbb{E}\left\{\sum_{r \in R} X_r\right\}$$

Solution concept

- ## Structure of optimal control laws

> Can we restrict attention to a subset of control laws without loosing optimality?

Examples: Markov policies in MDPs, linear policies in LQG, threshold policies in detection, etc.

Solution concept

- Structure of optimal control laws

  Can we restrict attention to a subset of control laws without loosing optimality?

  Examples: Markov policies in MDPs, linear policies in LQG, threshold policies in detection, etc.

- Sequential decomposition

  Can we pick the control laws one by one, instead of choosing all at once.

  Example: Dynamic programming

Sequential team form

A decentralized control system where the measurable spaces and the stochastic kernels are not specified

Sequential team form

A decentralized control system where the measurable spaces and the stochastic kernels are not specified

- System variables, control variables, stochastic variables (N, A, M, R)
- Information sets $\{I_k\}_{k \in N}$
  - Information structure $\{I_\alpha\}_{\alpha \in A}$
  - Dynamical coupling $\{I_m\}_{m \in M}$

Equivalence between team forms

Two team forms $\mathcal{T} = (N, A, M, R, \{I_k\}_{k \in N})$ and $\mathcal{T}^{'} = (N^{'}, A^{'}, M^{'}, R^{'}, \{I_k^{'}\}_{k \in N})$ are equivalent if

1. $N = N^{'}$, $A = A^{'}$, $M = M^{'}$, and $R = R^{'}$.
2. for all $m \in M$, $I_m = I_m^{'}$.
3. for any choice of measurable spaces $(\mathcal{X}_k, \mathfrak{F}_k)_{k \in N}$ and stochastic kernels $\{p_m\}_{m \in M}$, the value (optimal reward) of the teams corresponding to $\mathcal{T}$ and $\mathcal{T}^{'}$ are the same.

The first two conditions can be checked easily. There is no easy way to check the last condition.

Simplification of sequential teams

A team form $\mathcal{T}' = (N', A', M', R', \{I'_k\}_{k \in N})$ is a simplification of a team form $\mathcal{T} = (N, A, M, R, \{I_k\}_{k \in N})$ if

1. $\mathcal{T}'$ is equivalent to $\mathcal{T}$
2. $\displaystyle\sum_{\alpha \in A} |I'_k| < \sum_{\alpha \in A} |I_k|$

Simplification of sequential teams

A team form $\mathcal{T}^{'} = (N^{'}, A^{'}, M^{'}, R^{'}, \{I^{'}_k\}_{k \in N})$ is a simplification of a team form $\mathcal{T} = (N, A, M, R, \{I_k\}_{k \in N})$ if

1. $\mathcal{T}^{'}$ is equivalent to $\mathcal{T}$
2. $\displaystyle\sum_{\alpha \in A} |I^{'}_k| < \sum_{\alpha \in A} |I_k|$

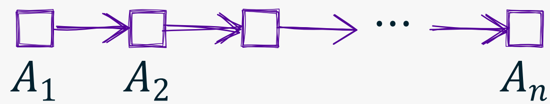
$\mathcal{T}^{'}$ is a strict simplification of $\mathcal{T}$ if

1. $\mathcal{T}^{'}$ is equivalent to $\mathcal{T}$
2. $|I^{'}_\alpha| \leq |I_\alpha|$, $\alpha \in A$, and at least one of the inequalities is strict.

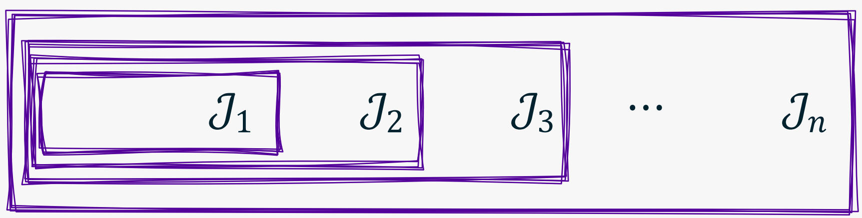
Given a team form, can we simplify it?

Centralized stochastic control

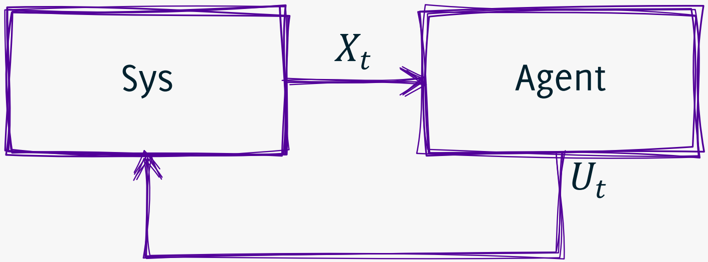
Single decision maker



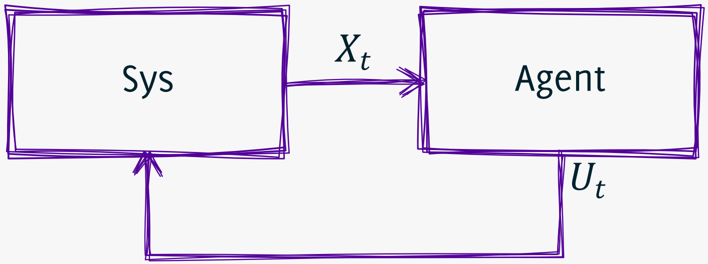
with classical information structures



MDP: Structural properties



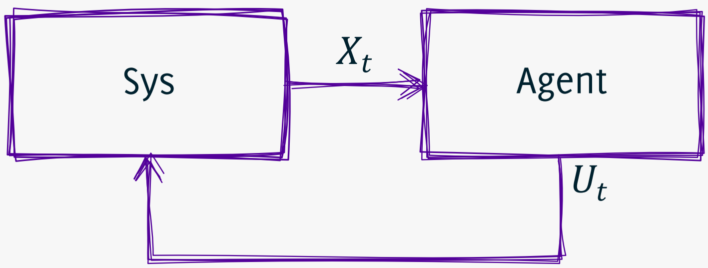
MDP: Structural properties

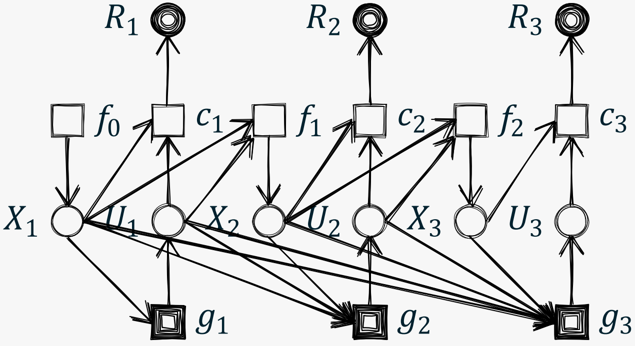


Structure of optimal policy

Choose current action based on current state $\mathbf{X_t}$

MDP: Structural properties

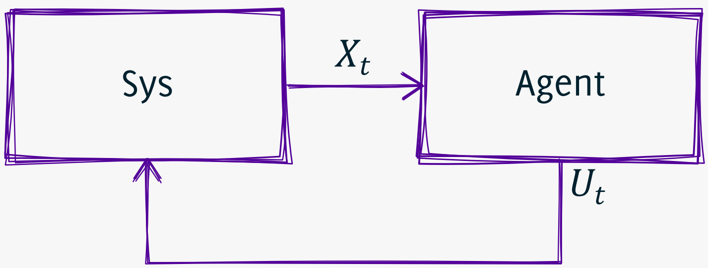


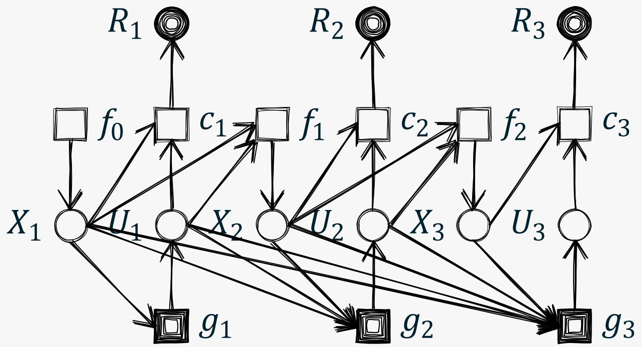


Structure of optimal policy

Choose current action
based on current state $\mathbf{X_t}$

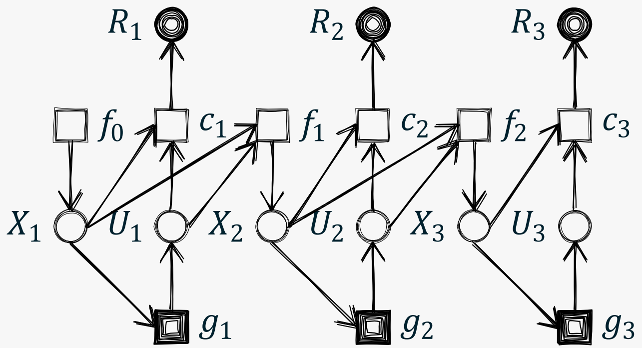
MDP: Structural properties



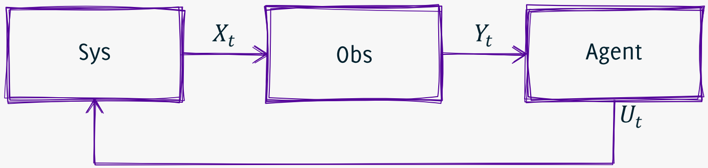


Structure of optimal policy

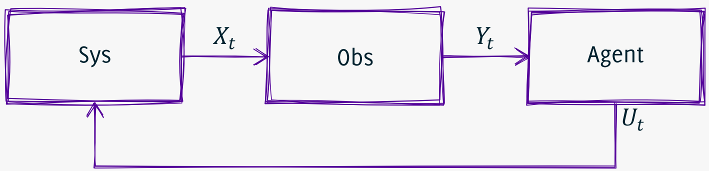
Choose current action
based on current state $\mathbf{X_t}$



POMDP: Structural properties



POMDP: Structural properties

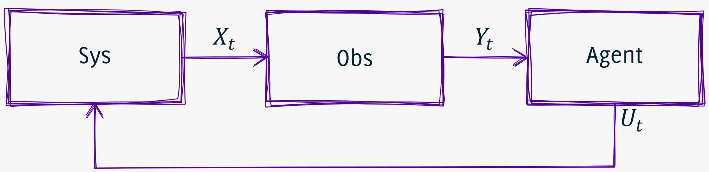


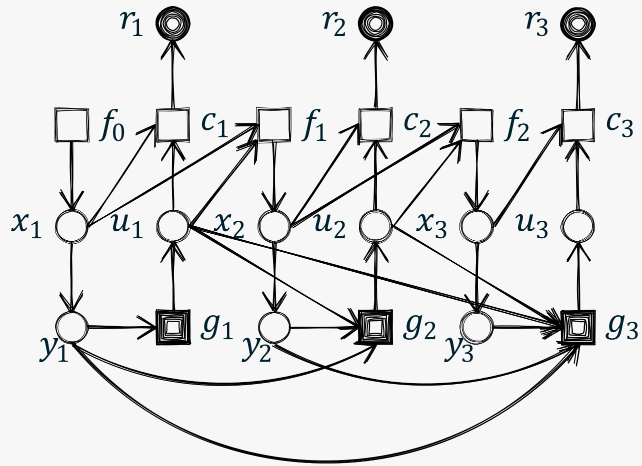
Structure of optimal policies

Choose current action
based on current info state

Pr(state of system | all data at agent)

POMDP: Structural properties



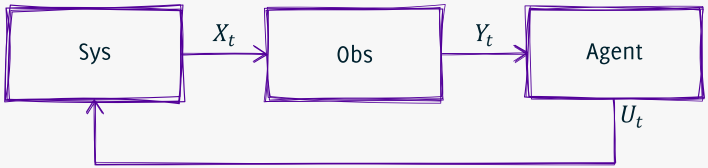


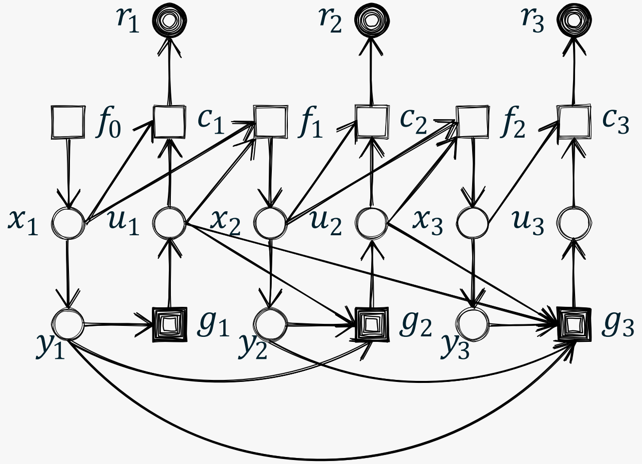
Structure of optimal policies

Choose current action
based on current info state

Pr(state of system | all data at agent)

POMDP: Structural properties

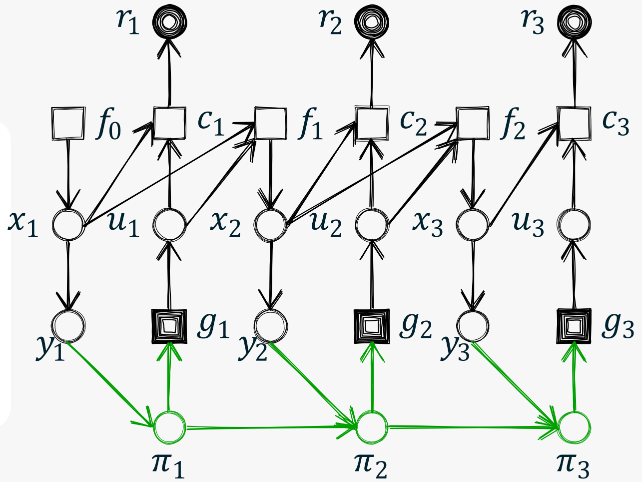




Structure of optimal policies

Choose current action
based on current info state

Pr(state of system | all data at agent)



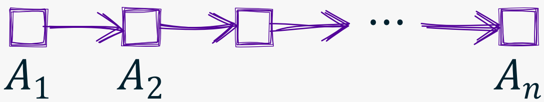
Structural policies in stochastic control

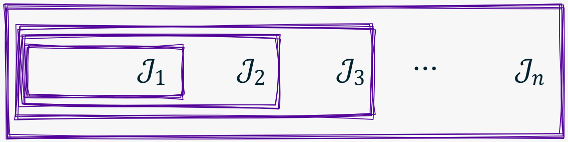
- Structure of optimal policies
  - Shed irrelevant information
  - Compress relevant information to a compact statistic
  - Hopefully, the data at the agent is not increasing with time

Structural policies in stochastic control

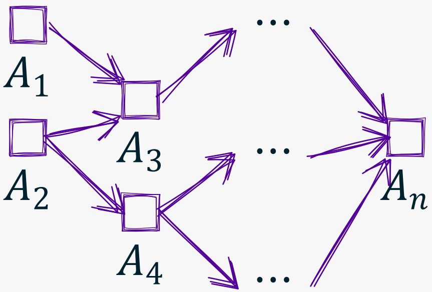
- Structure of optimal policies
  - Shed irrelevant information
  - Compress relevant information to a compact statistic
  - Hopefully, the data at the agent is not increasing with time
- Implication of the results
  - Simplify the functional form of the decision rules
  - Simplify search for optimal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

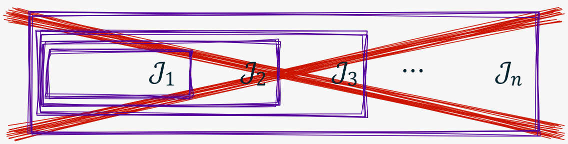
Extending ideas to decentralized control



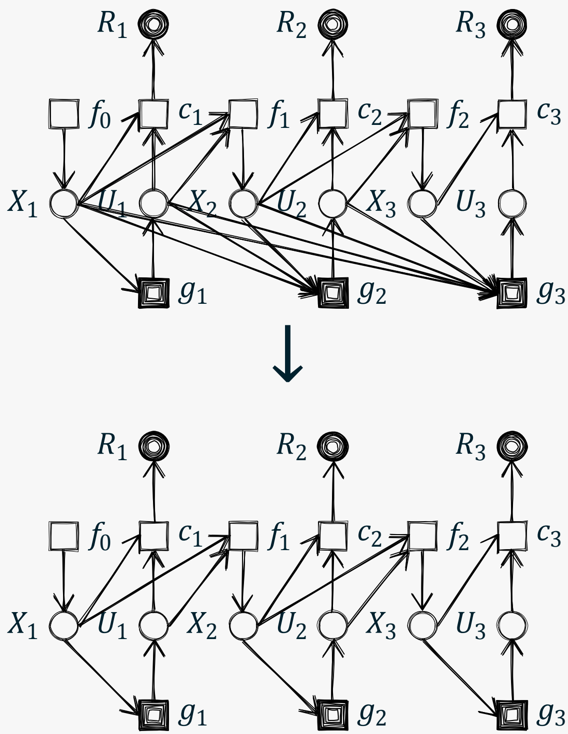


$\downarrow$





Can we generalize the reasoning of MDPs to decentralized systems



The textbook proof

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

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

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

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

The textbook proof

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

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

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

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

$W_t(x_t) = V_t(x_1, \dots, x_t)$ for all $x_1, \dots, x_t$

The textbook proof

De

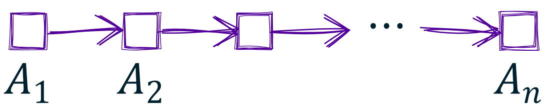
De

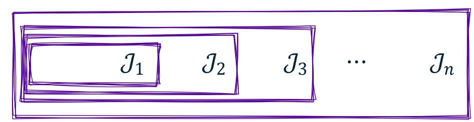
By

Re

Proof tied to

centralized system. . .

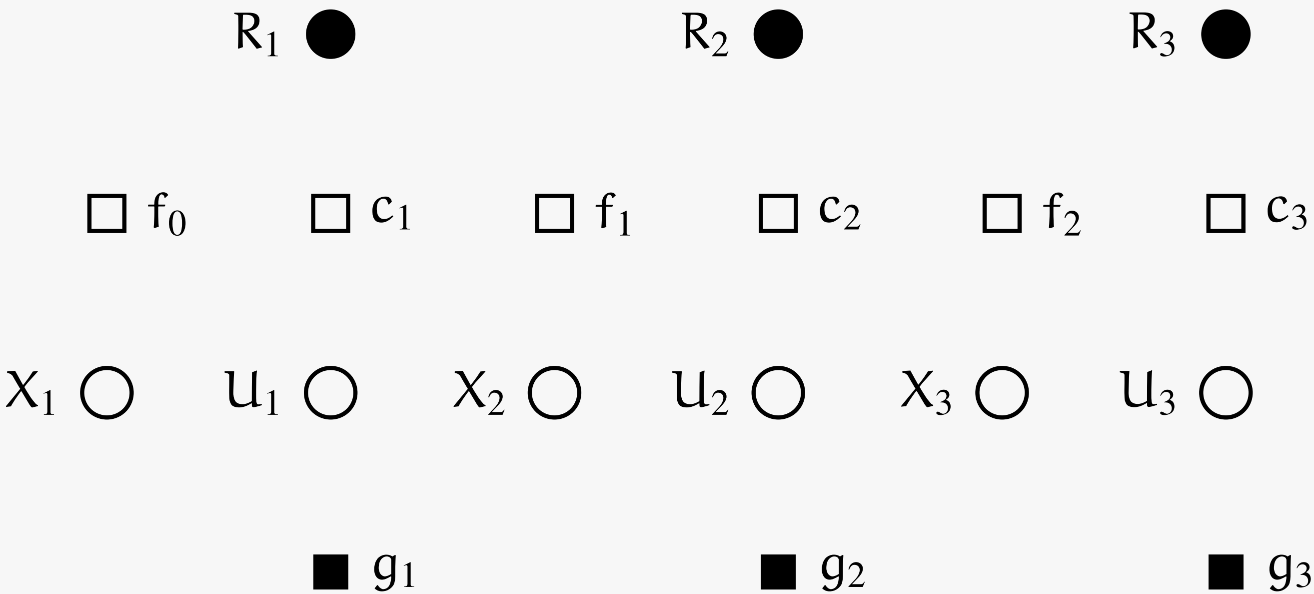




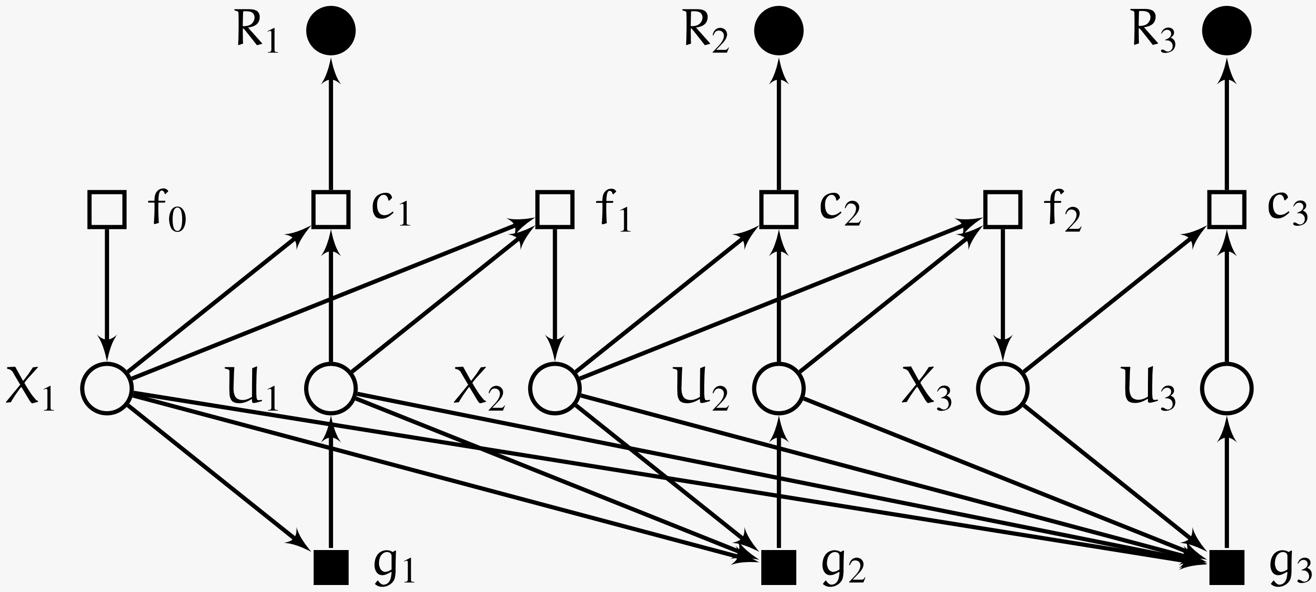
$W_t(x_t) = V_t(x_1, \ldots, x_t)$ for all $x_1, \ldots, x_t$

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

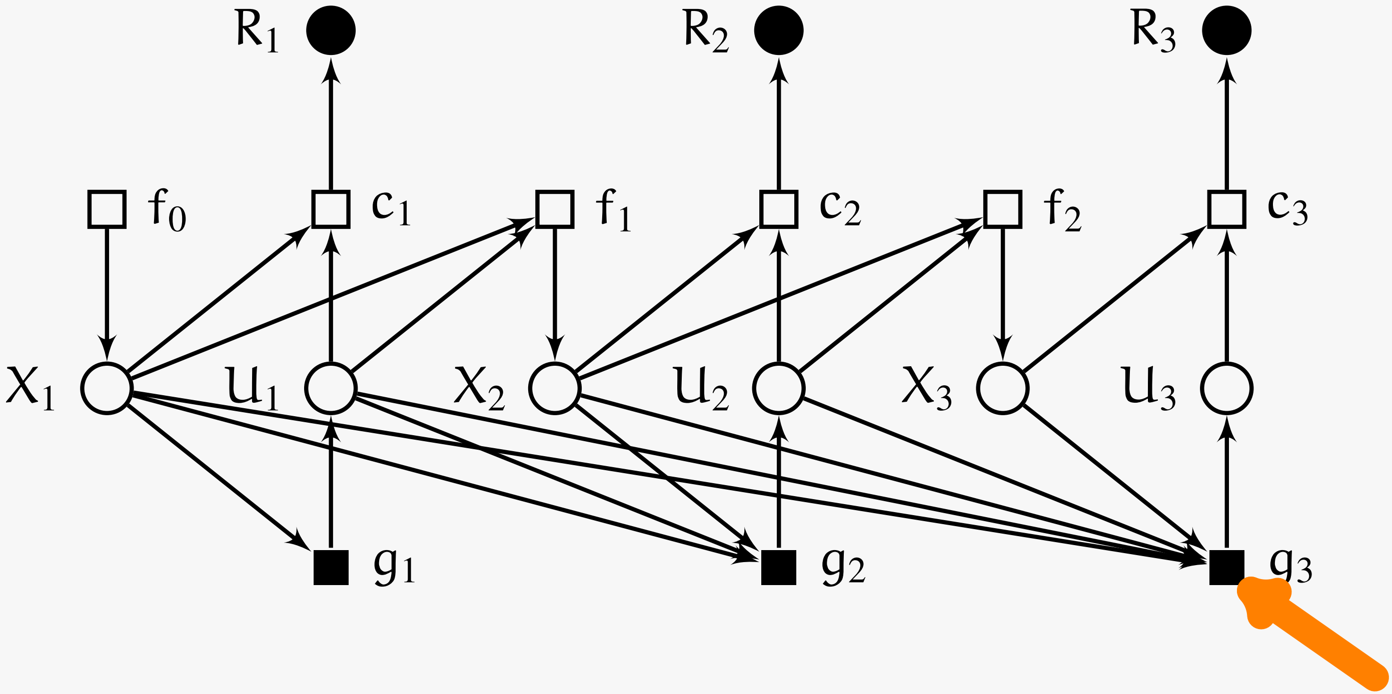
A graphical modeling proof



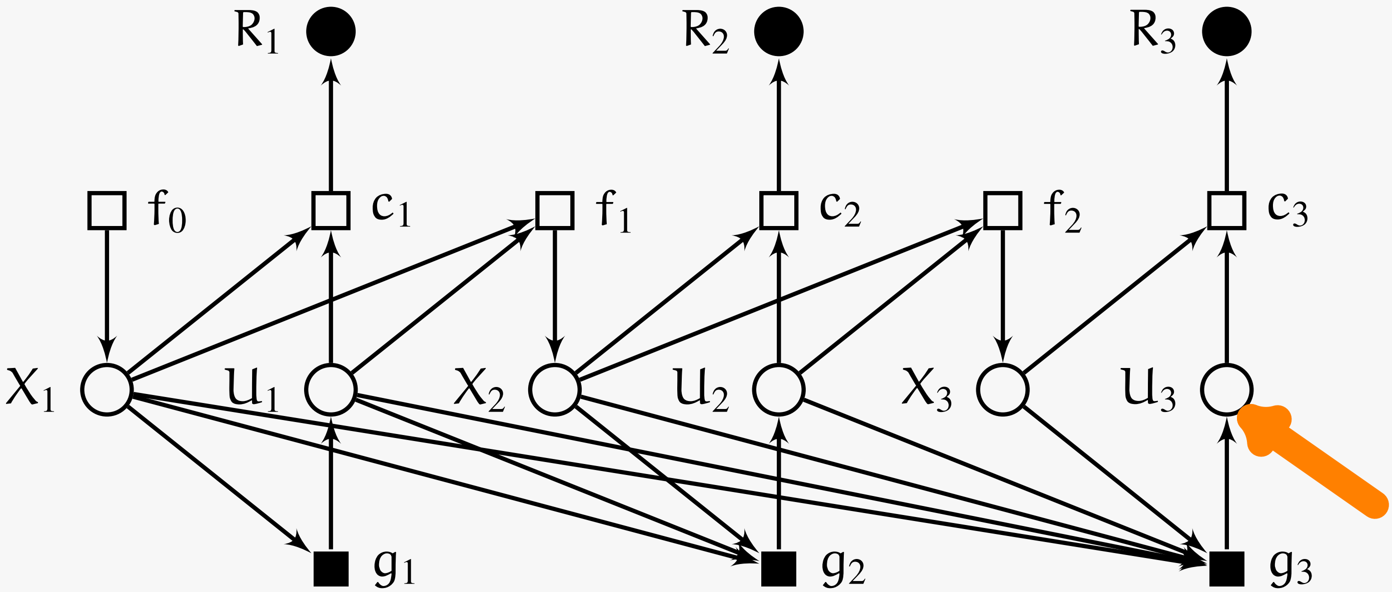
A graphical modeling proof



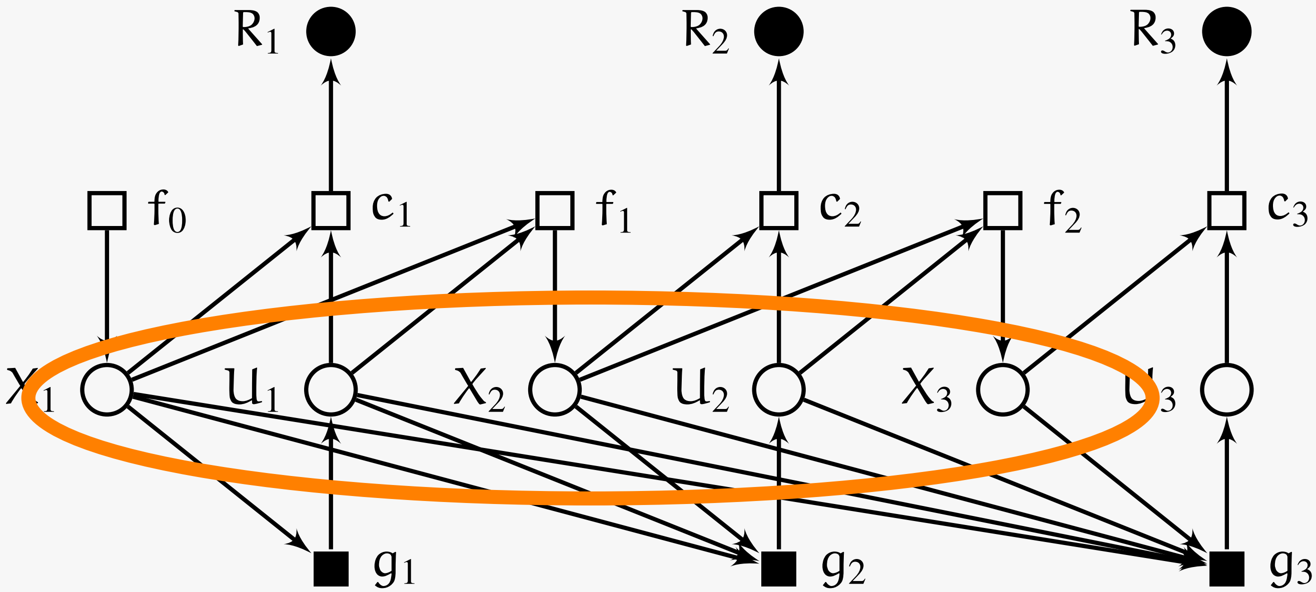
agent at time 3



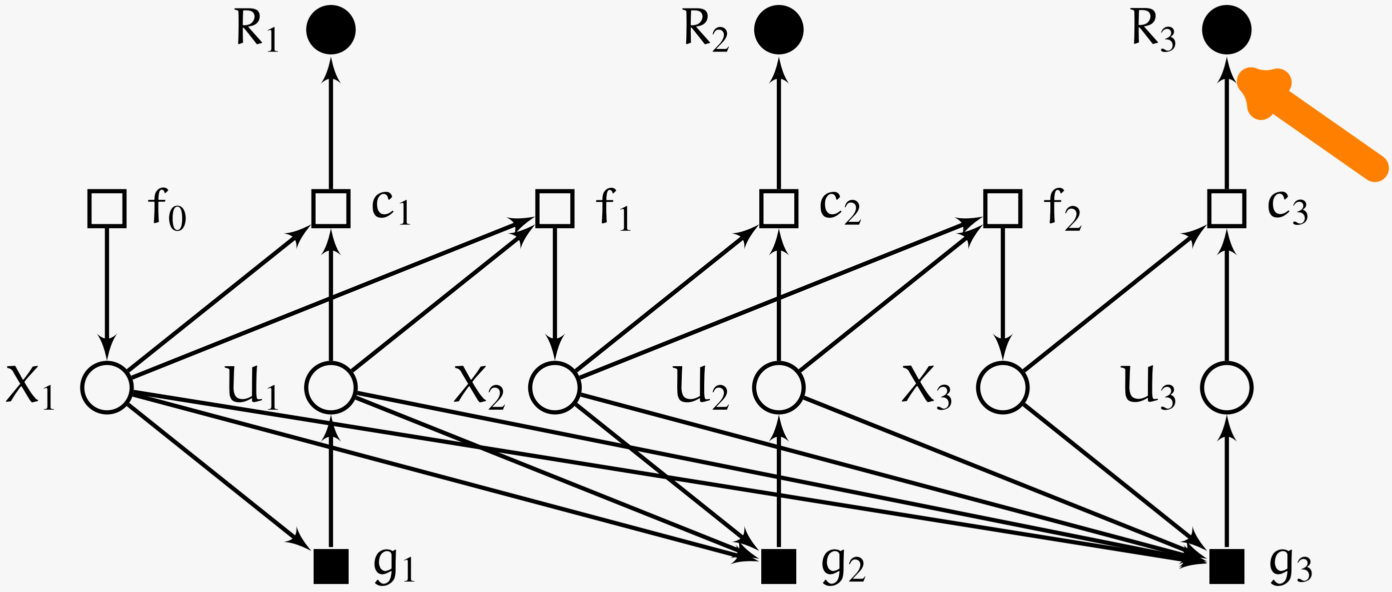
control action



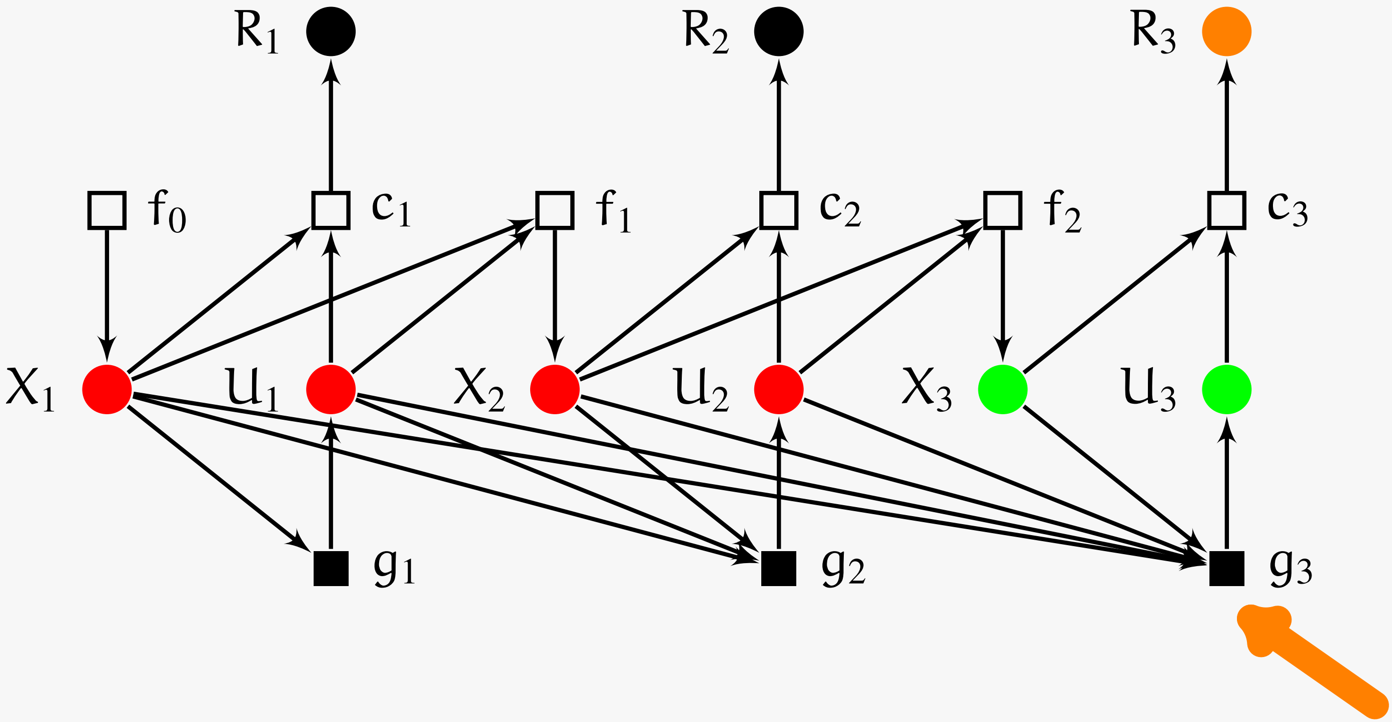
observations



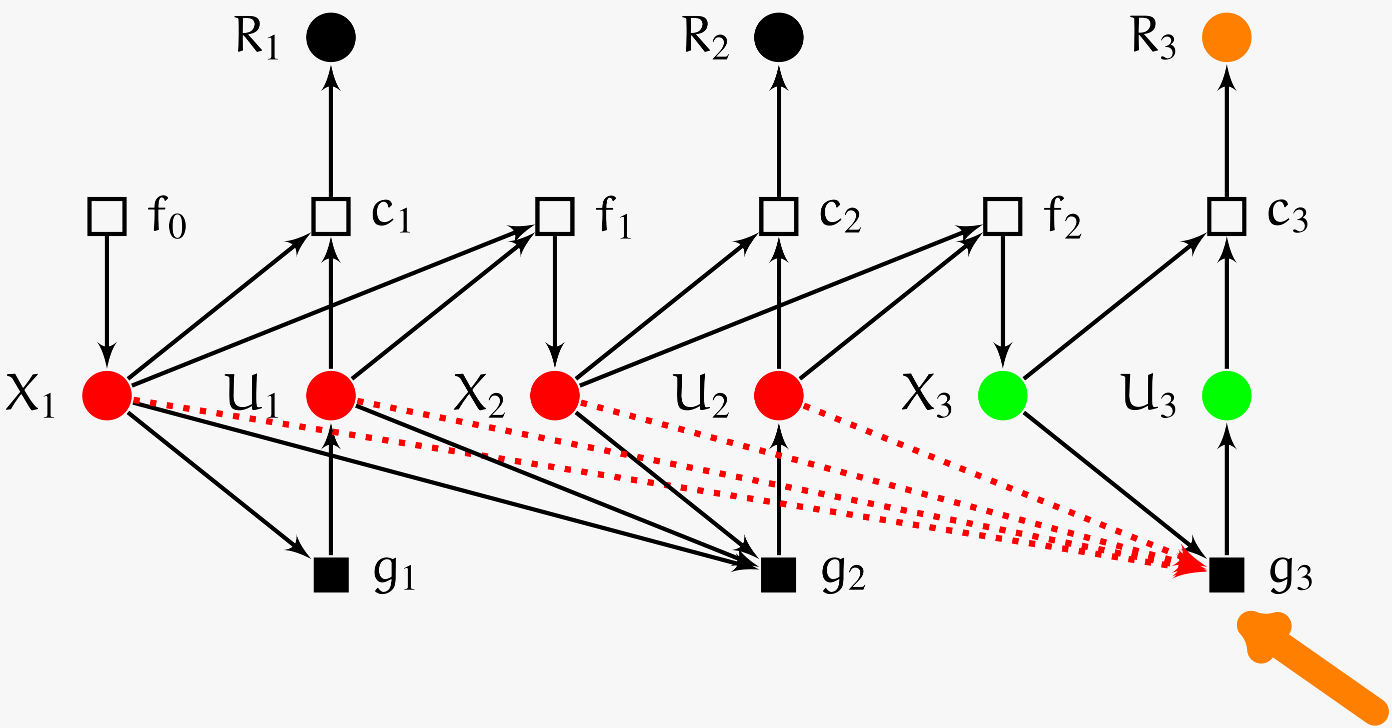
dependent reward



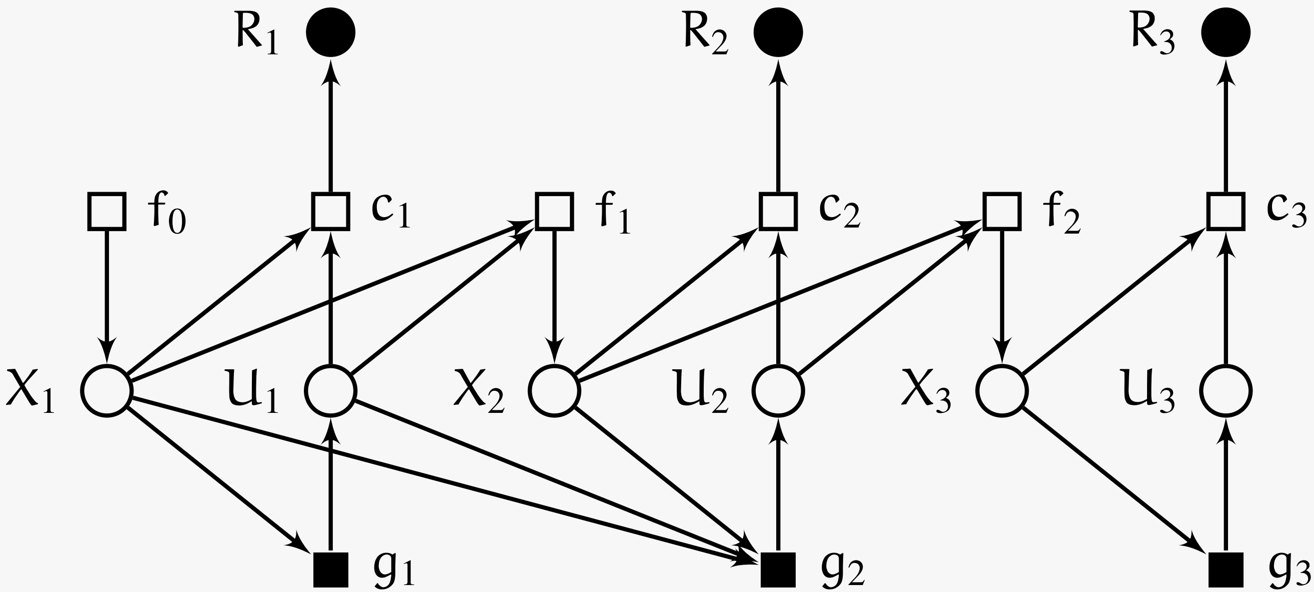
irrelevant observations



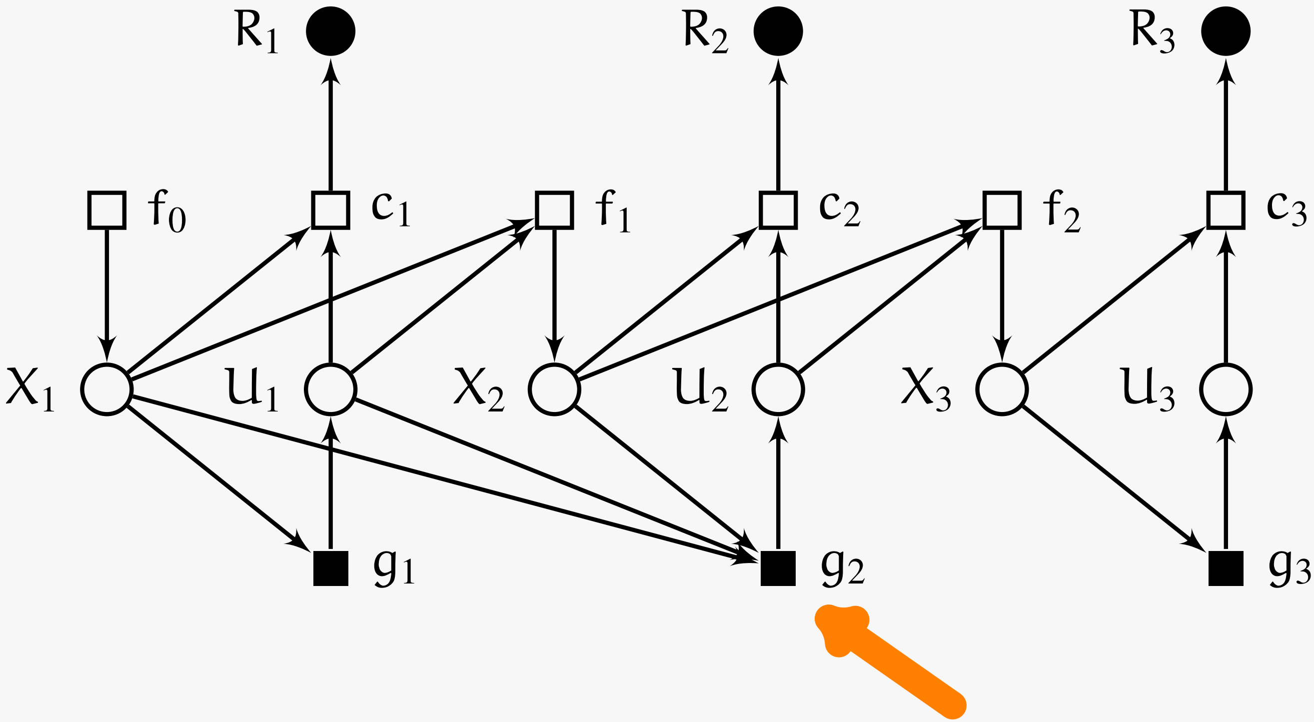
remove edges



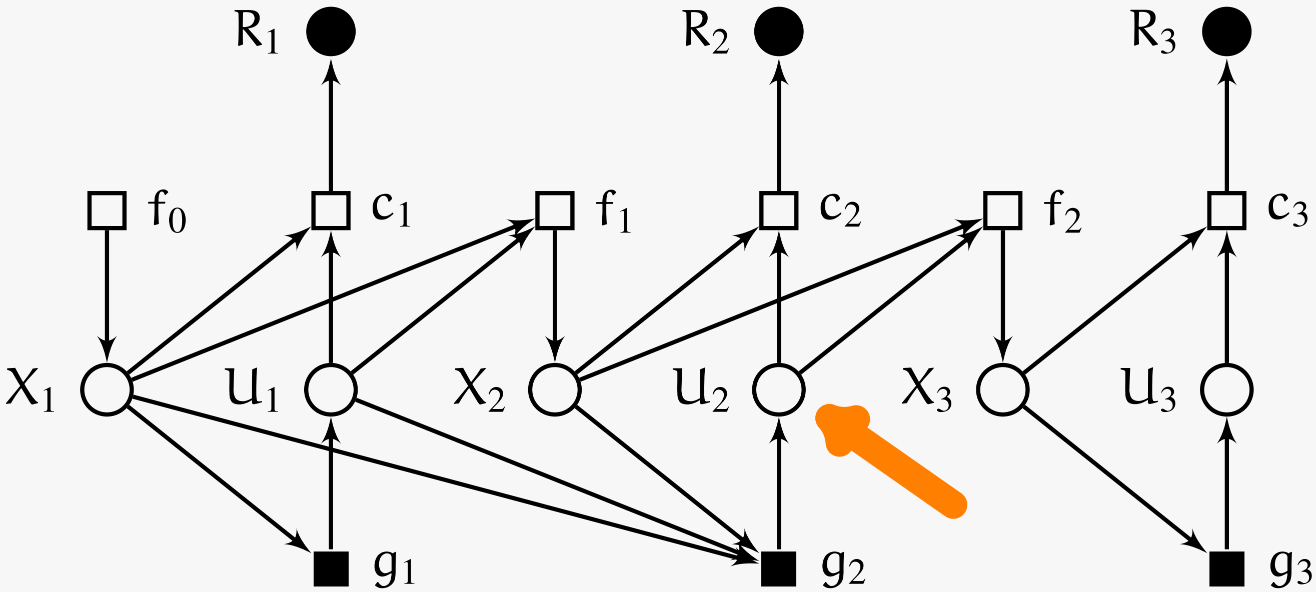
repeat



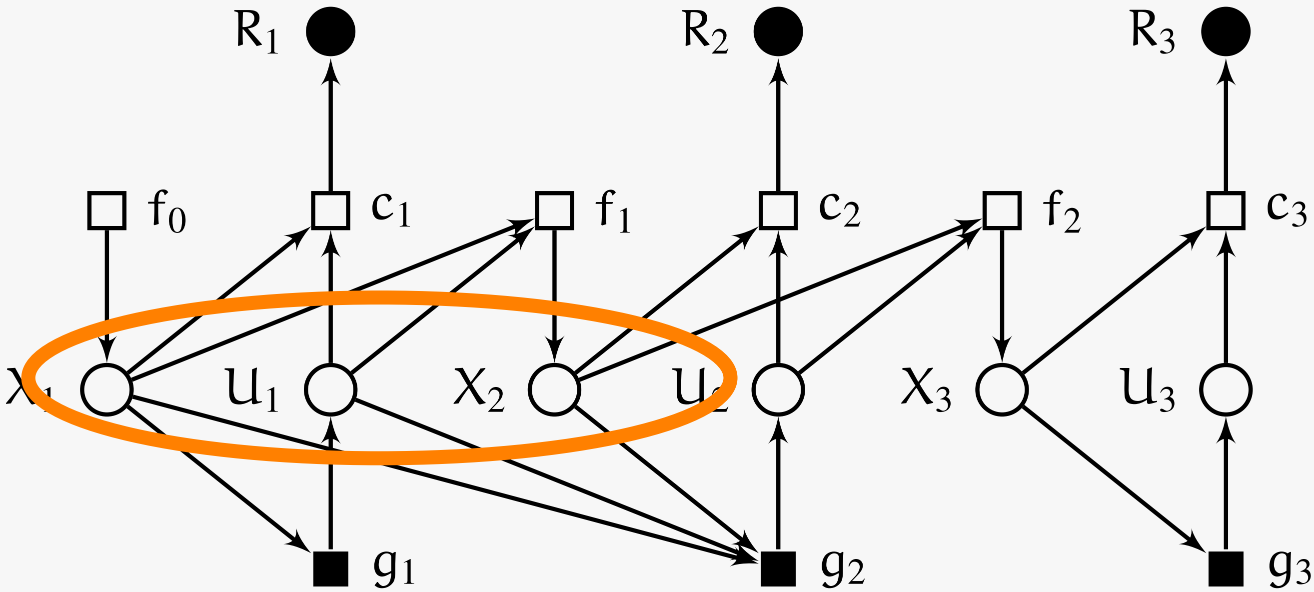
agent at time 2



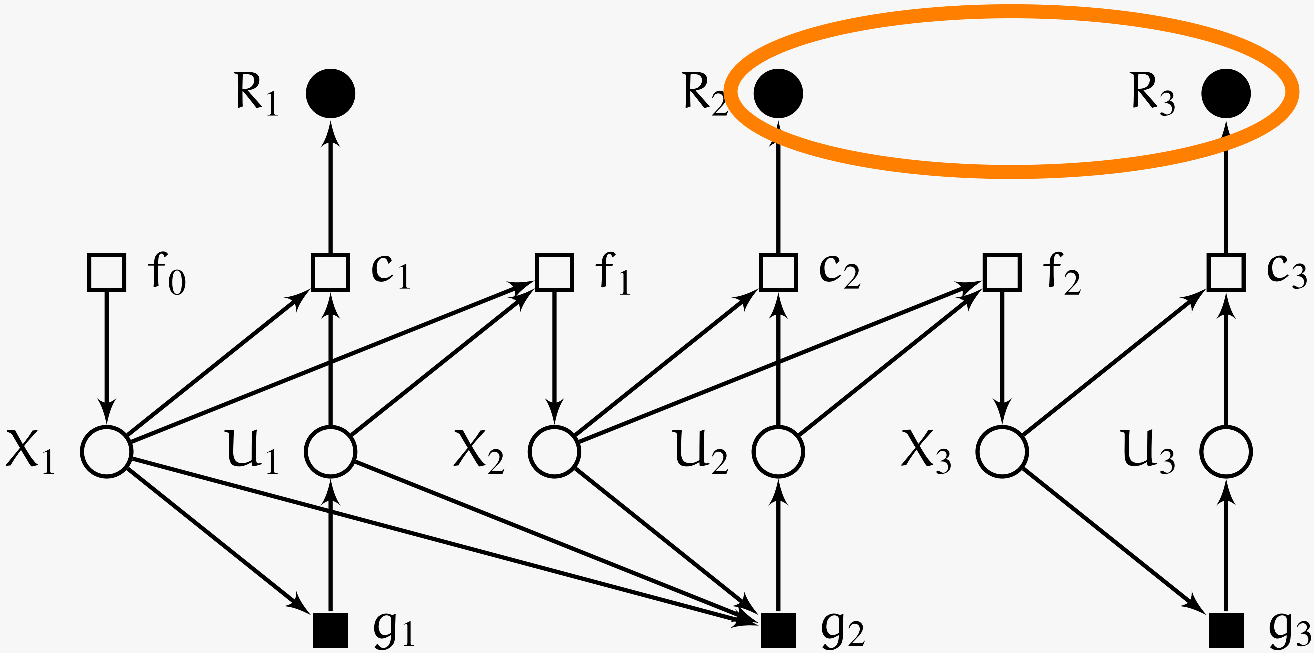
control action



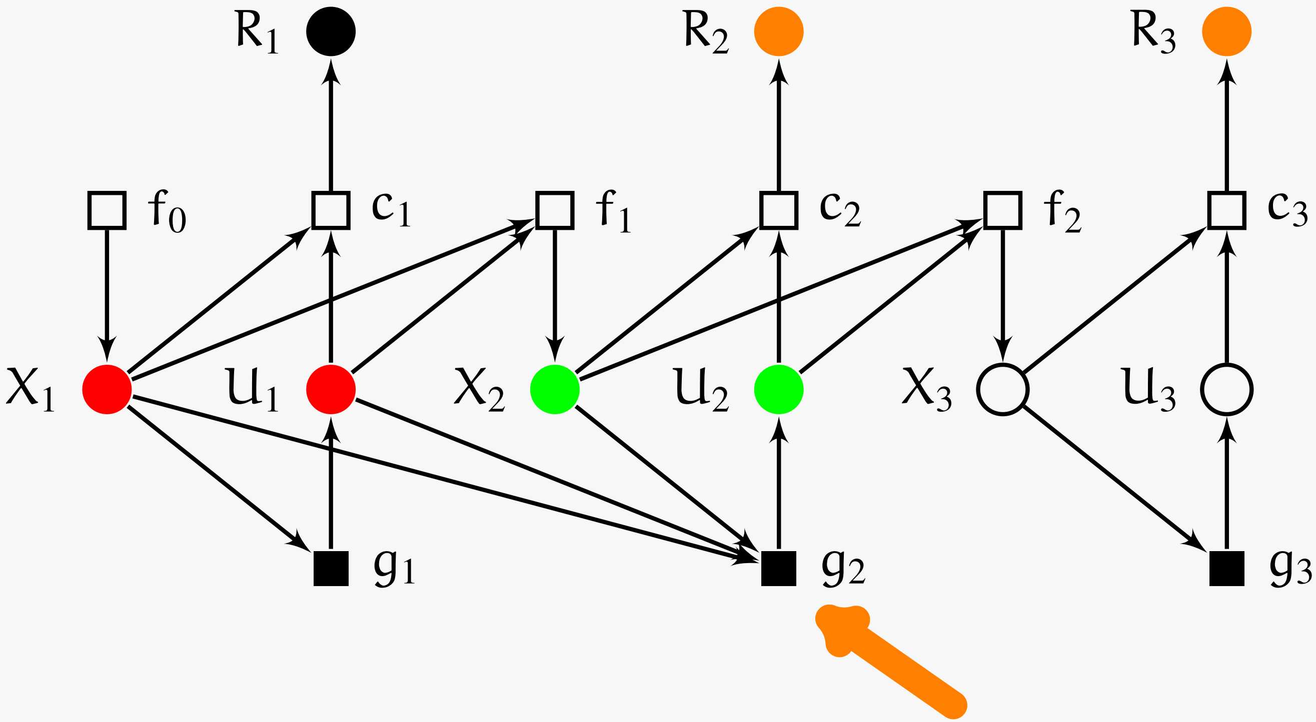
observations



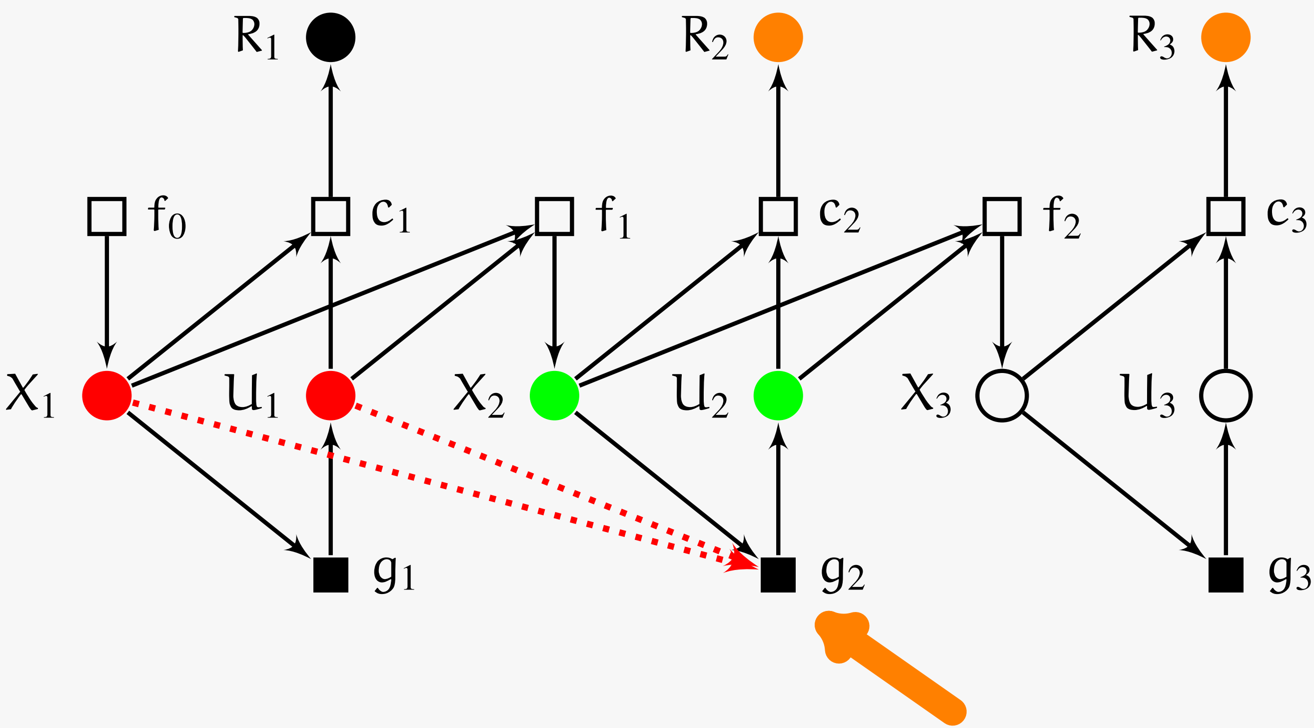
dependent rewards



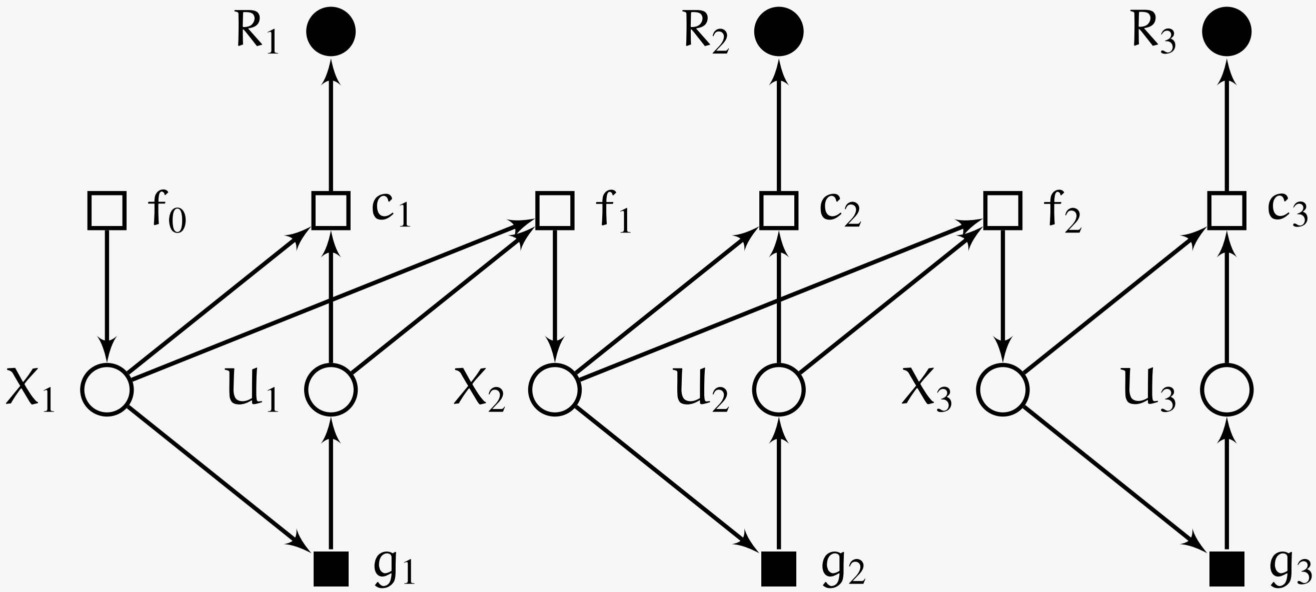
irrelevant observations



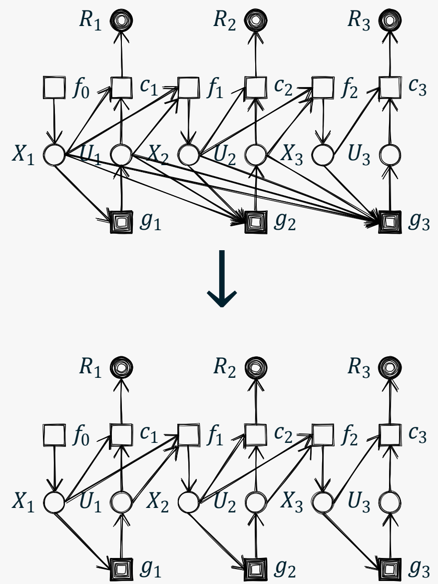
remove edges



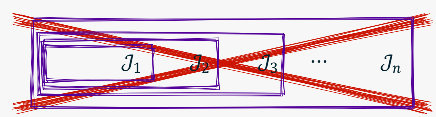
we are done



Shedding irrelevant information

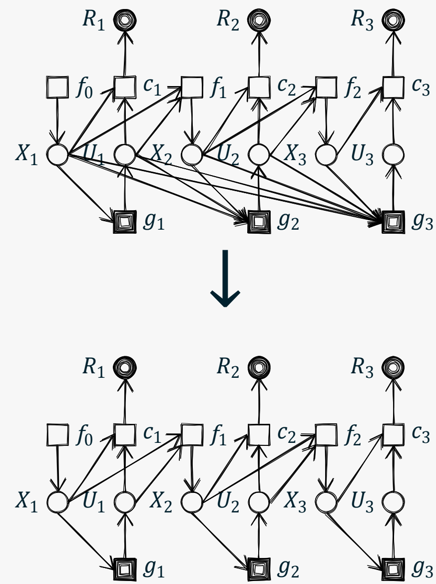


applied to

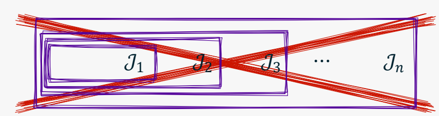


Shedding irrelevant information

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

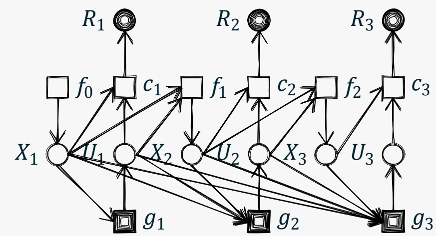


applied to

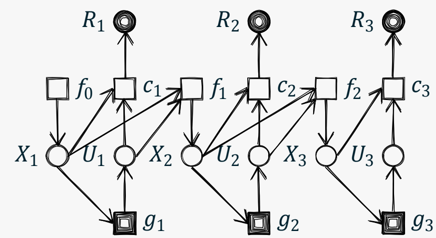


Shedding irrelevant information

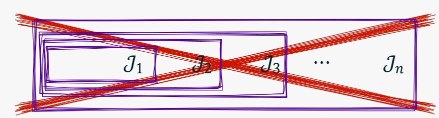
- Iterative procedure
  - Shed irrelevant data at an agent (at a particular time)
  - Iterate over all agents until a fixed point
  - Repeat for all coordinators of groups of agents



↓



applied to



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

- Coordinator, Common information, state for input-output mapping

Automating the procedure

- Irrelevant data, dependent rewards, conditional independence

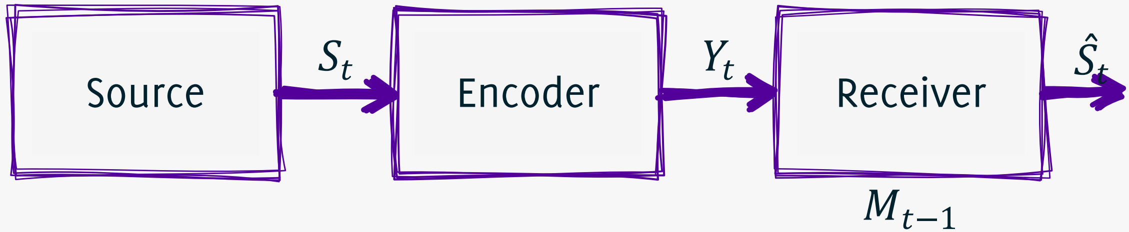
Directed acyclic graphs and graphical models

- Coordinator, 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, BSTJ-79



First order Markov source $\{S_t, t = 1, \ldots, T\}$.

Real-Time Encoder: $Y_t = c_t(S^t, Y^{t-1})$

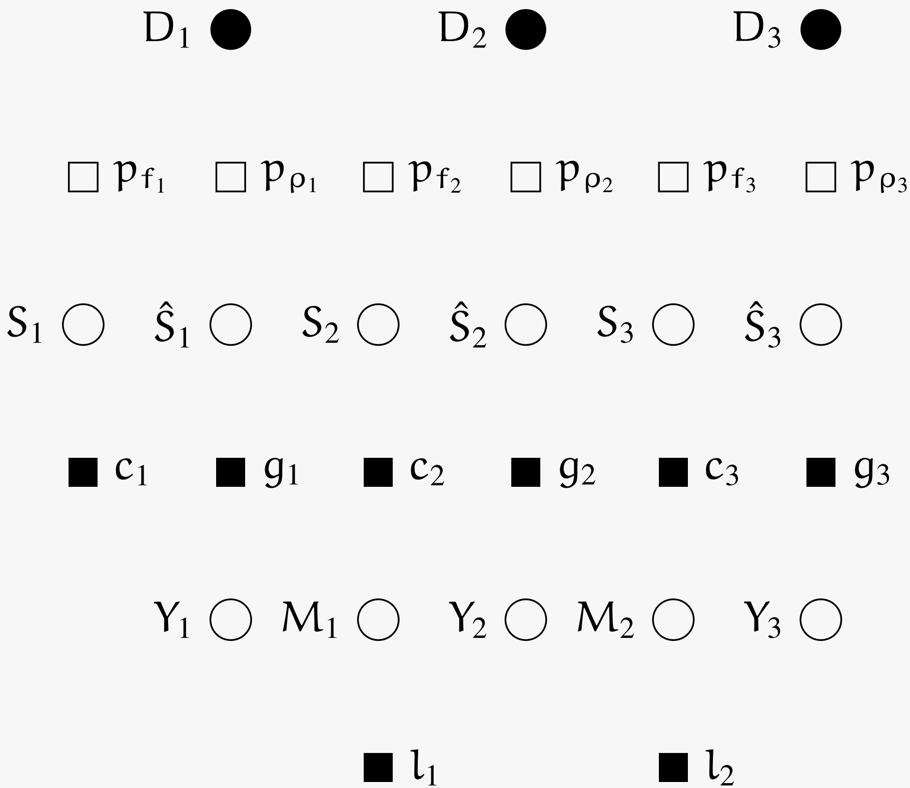
Real-Time Finite Memory Decoder: $\hat{S}_t = g_t(Y_t, M_{t-1})$

$M_t = l_t(Y_t, M_{t-1})$

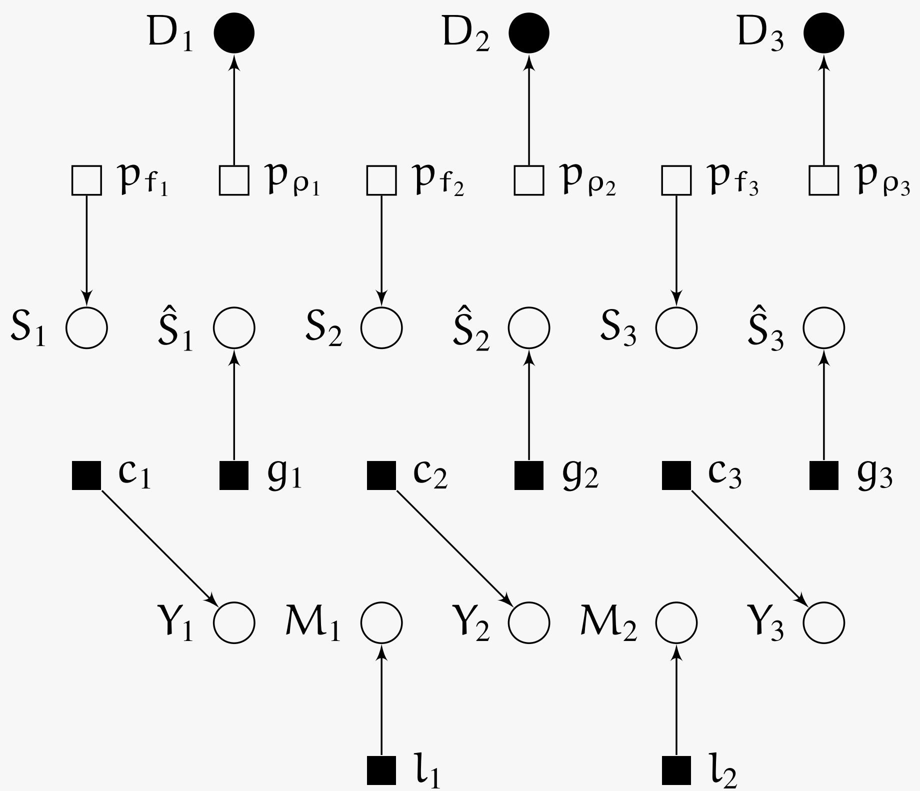
Instantaneous distortion $\rho(S_t, \hat{S}_t)$

Objective: minimize $E\left\{\sum_{t=1}^{T} \rho(S_t, \hat{S}_t)\right\}$

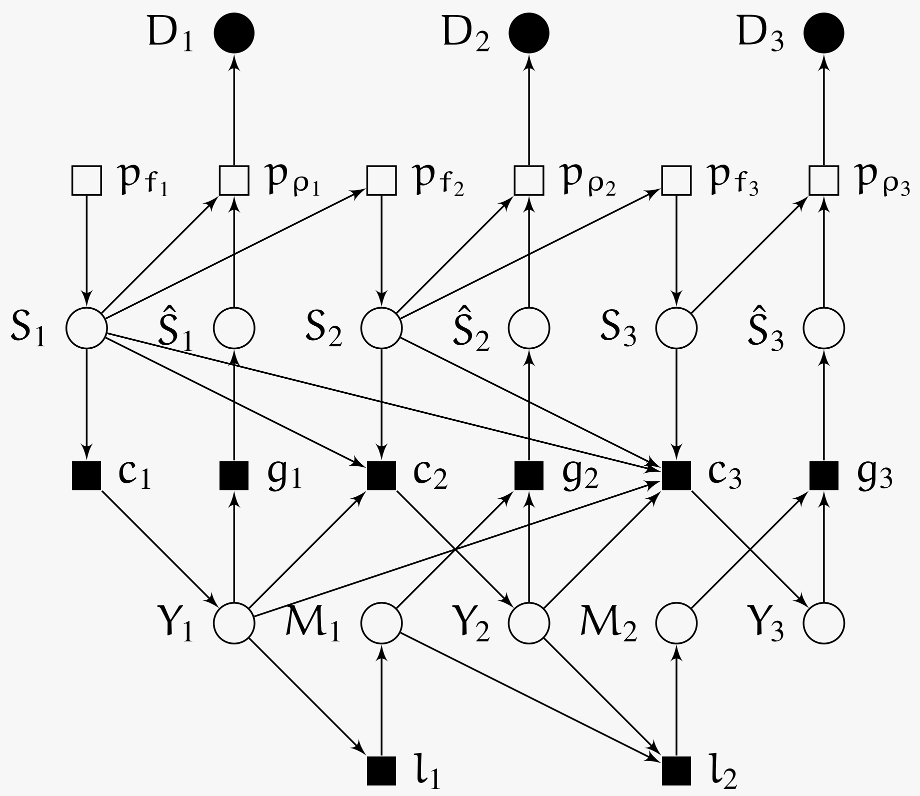
An Example: Real-time communication



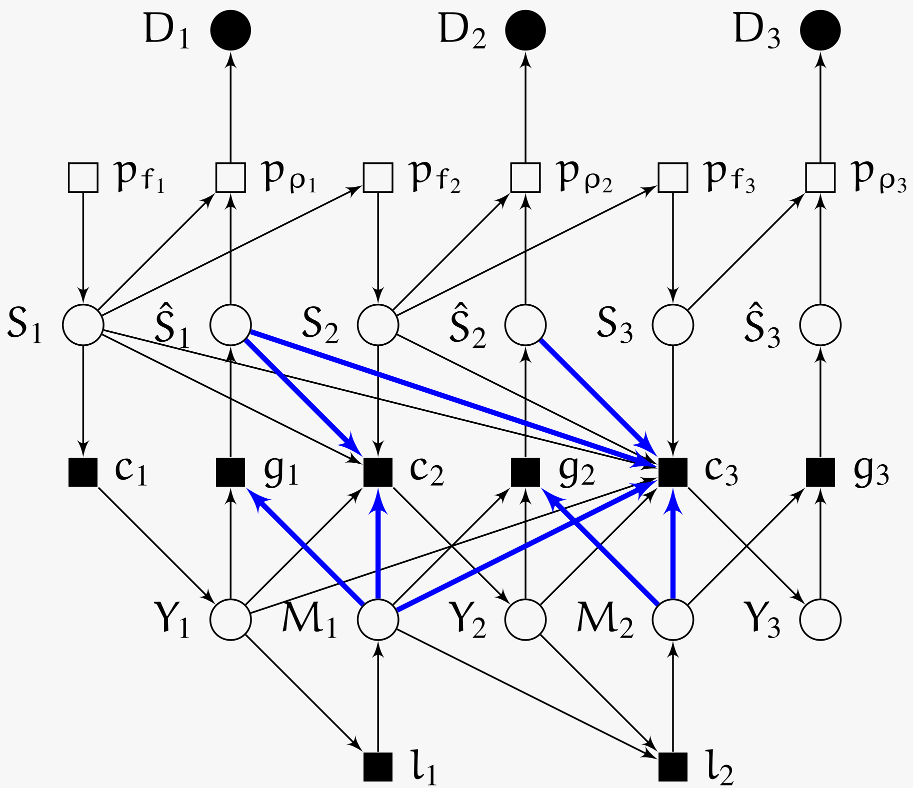
An Example: Real-time communication



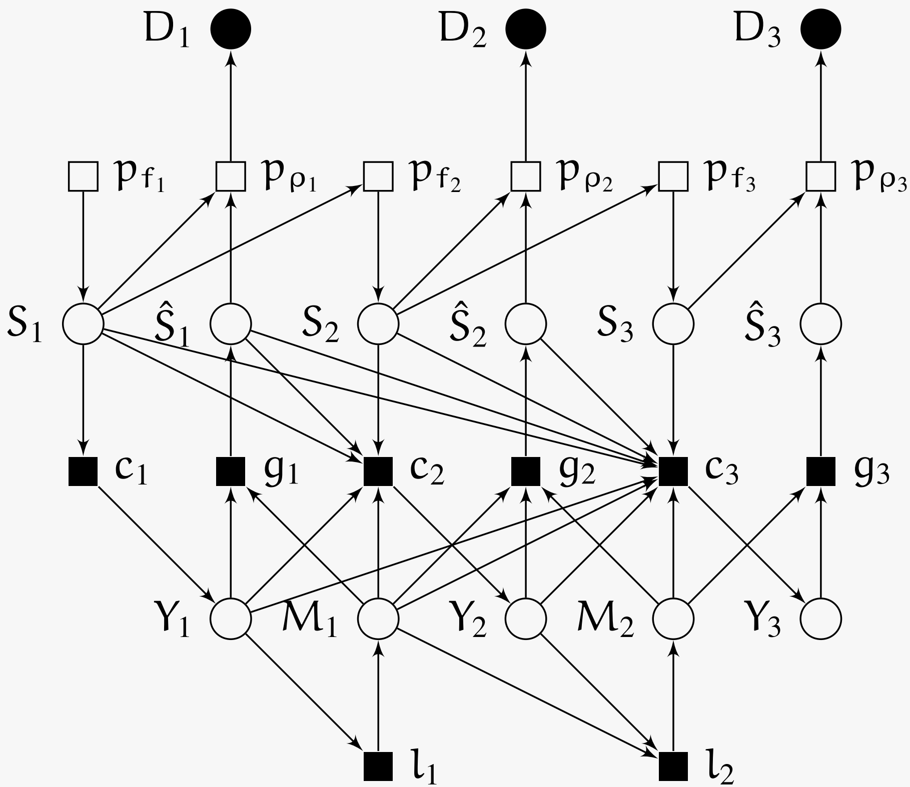
An Example: Real-time communication



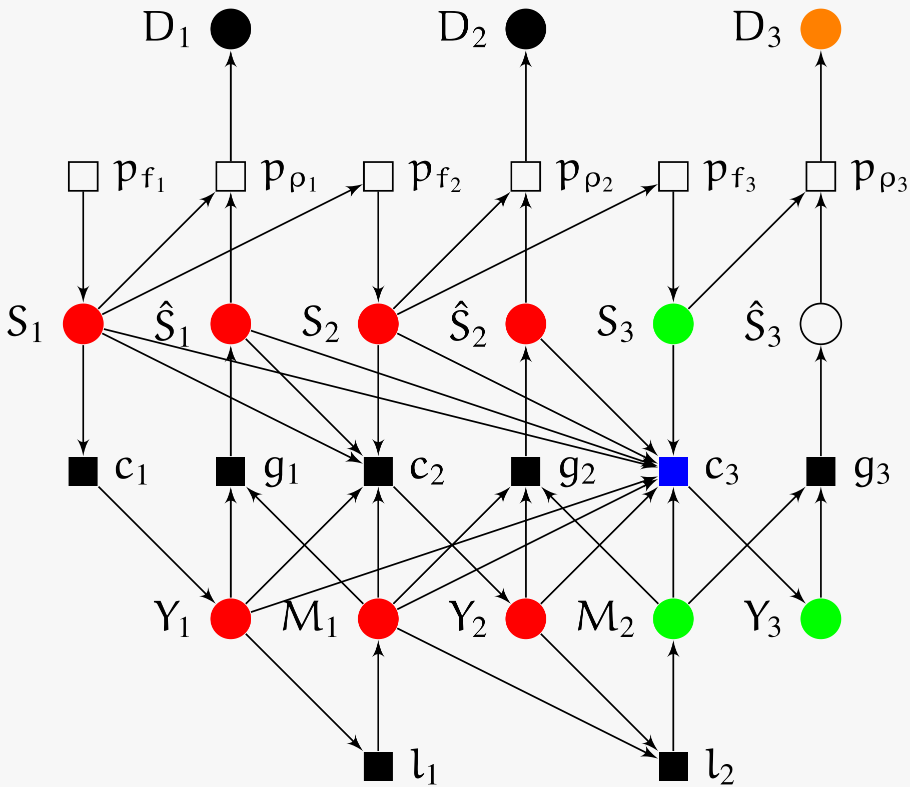
Completion of a team



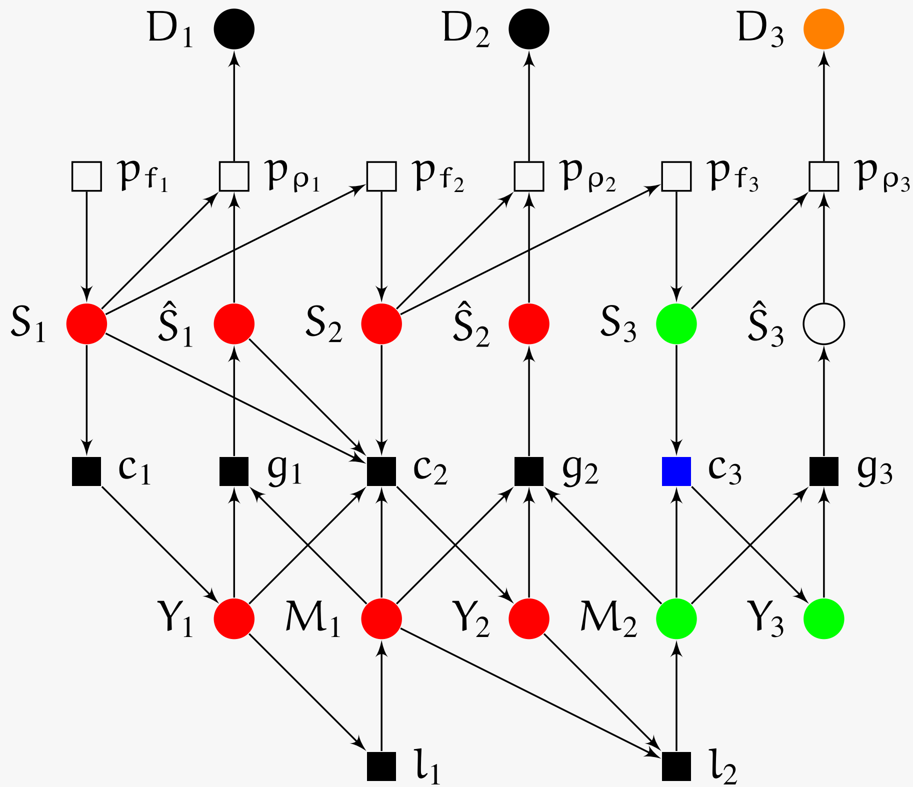
Remove irrelevant nodes



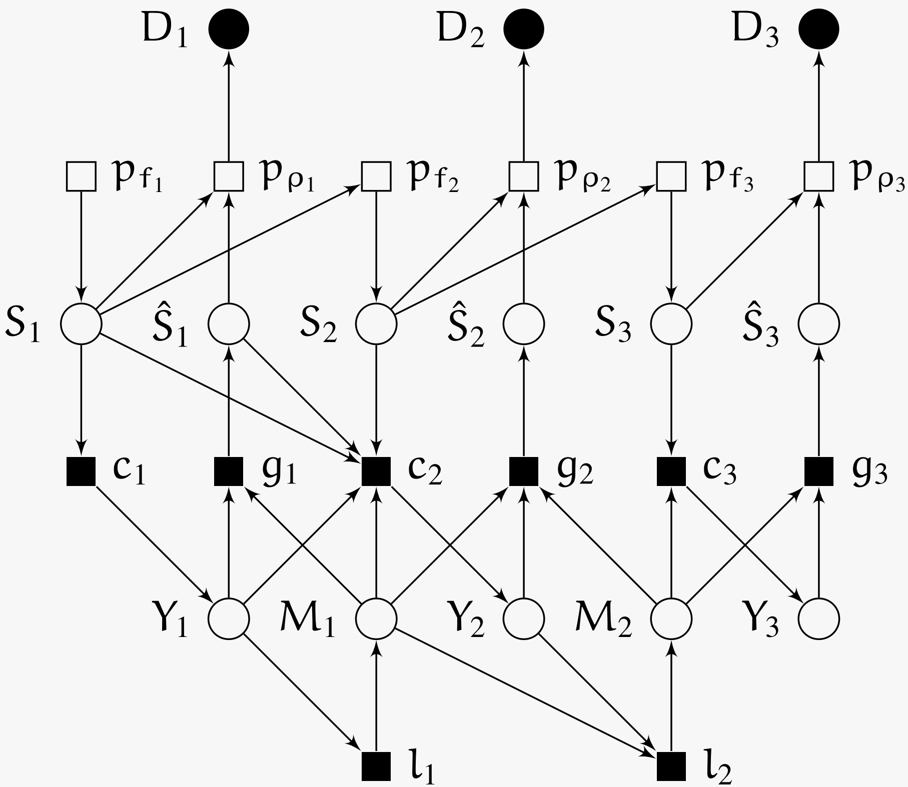
Remove irrelevant nodes



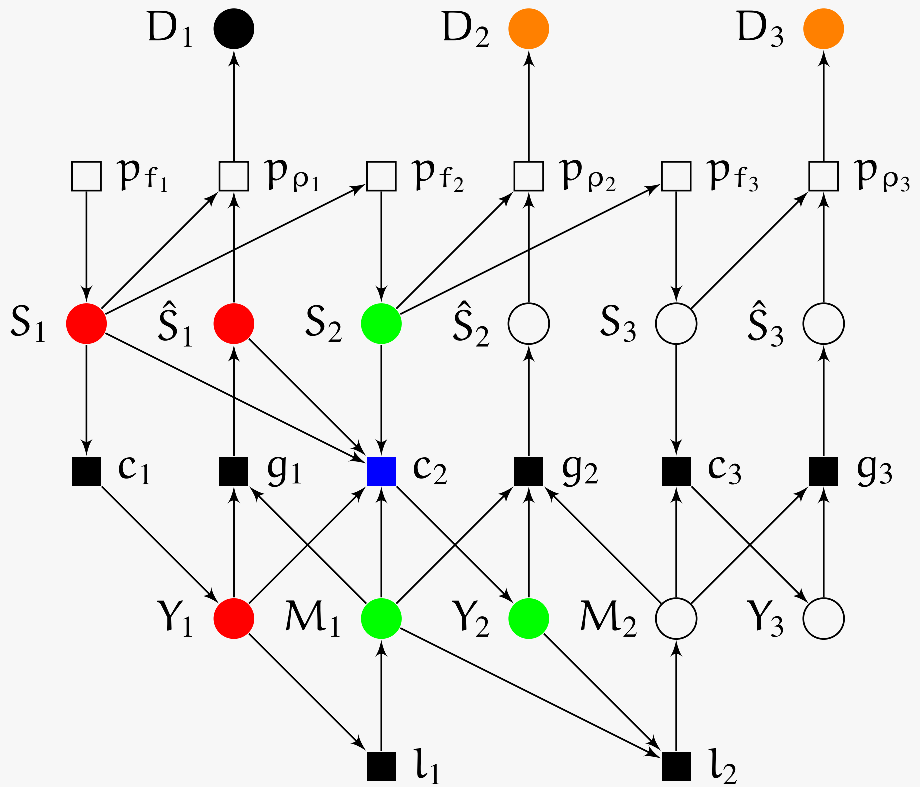
Remove irrelevant nodes



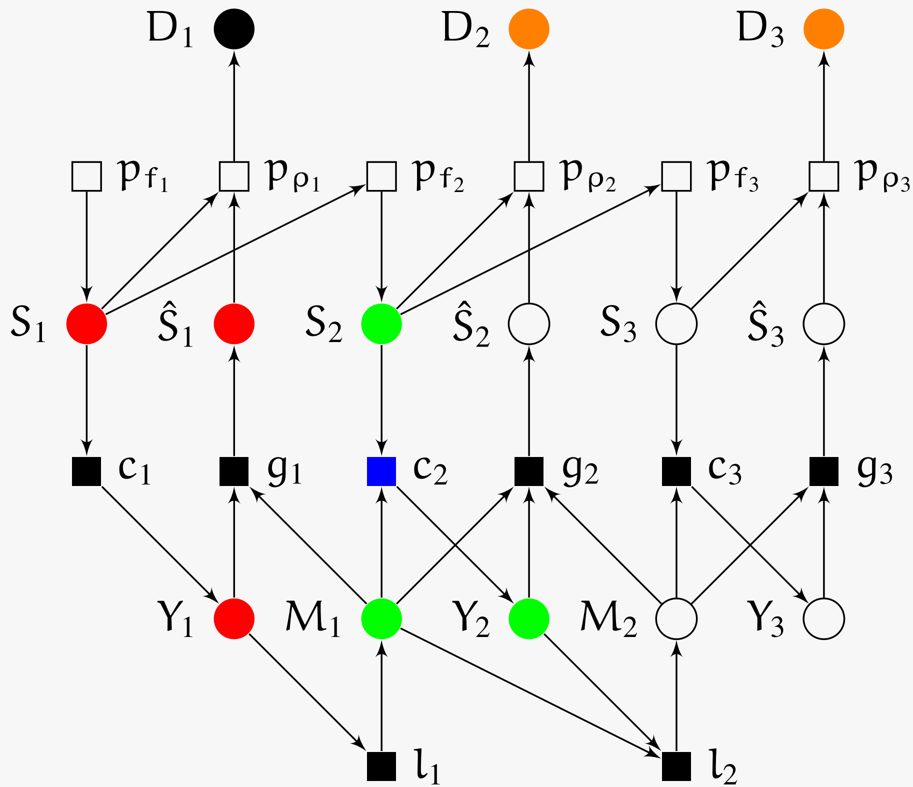
Remove irrelevant nodes



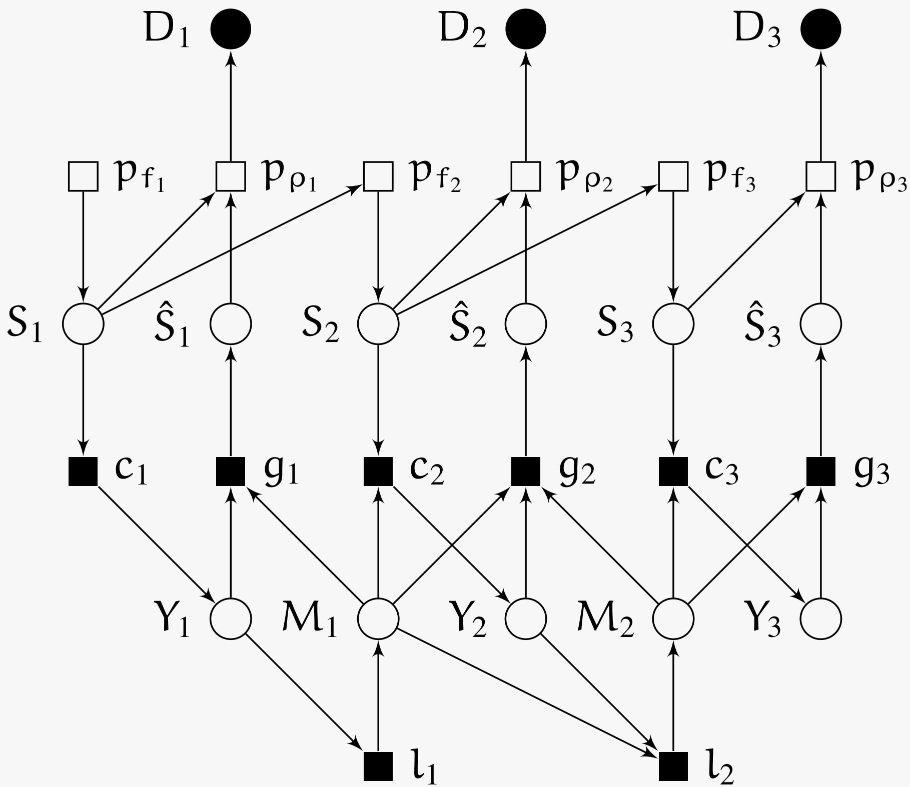
Remove irrelevant nodes



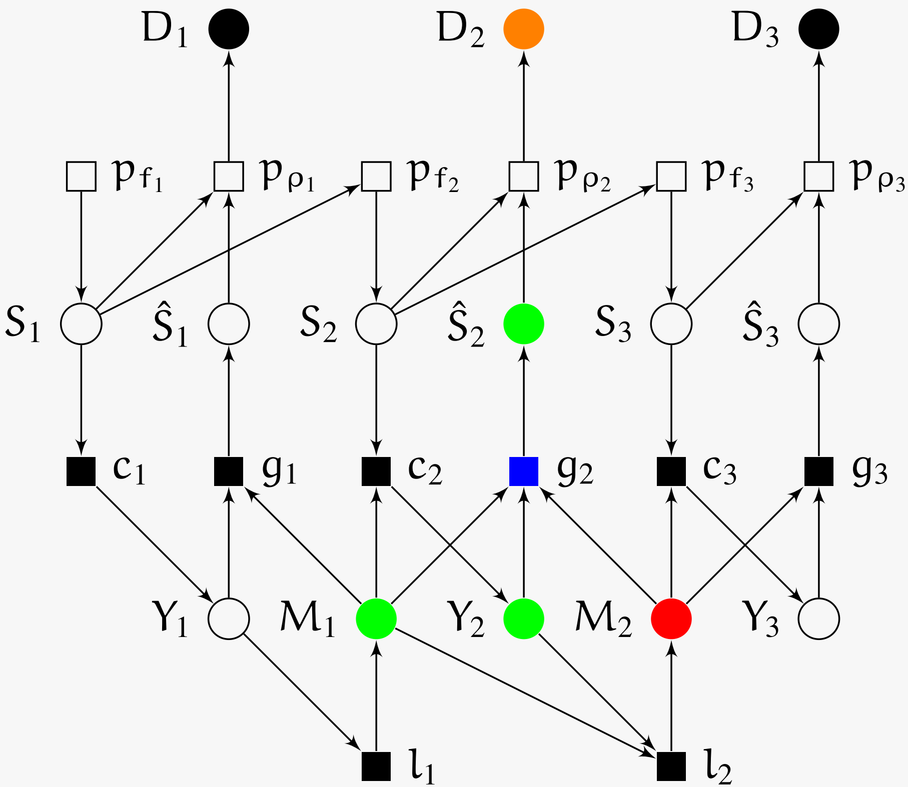
Remove irrelevant nodes



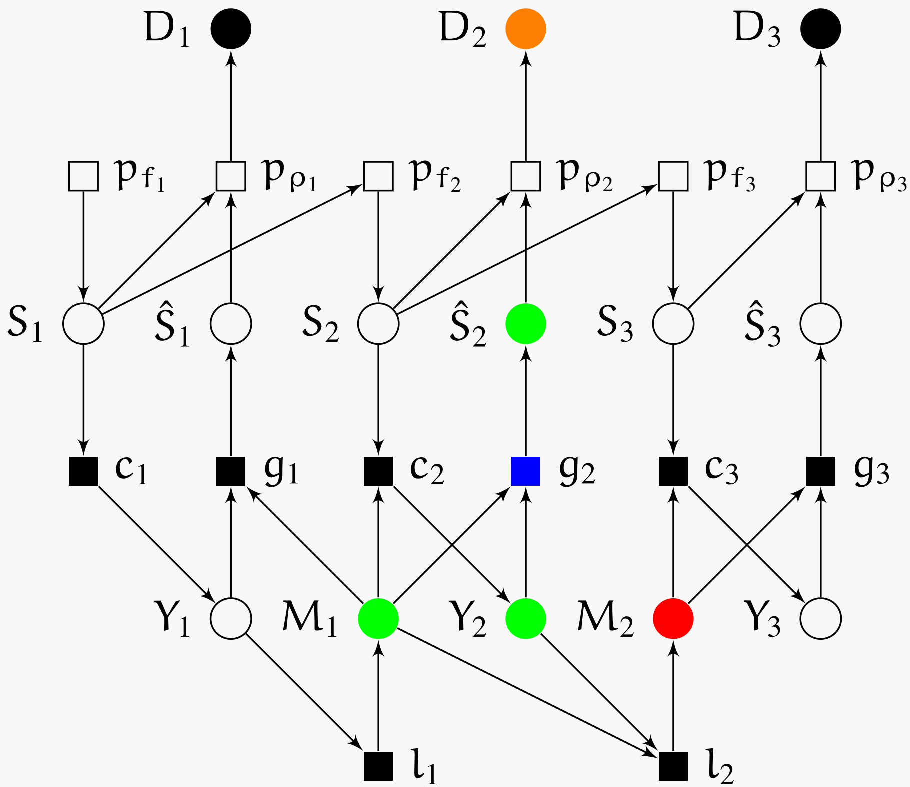
Remove irrelevant nodes



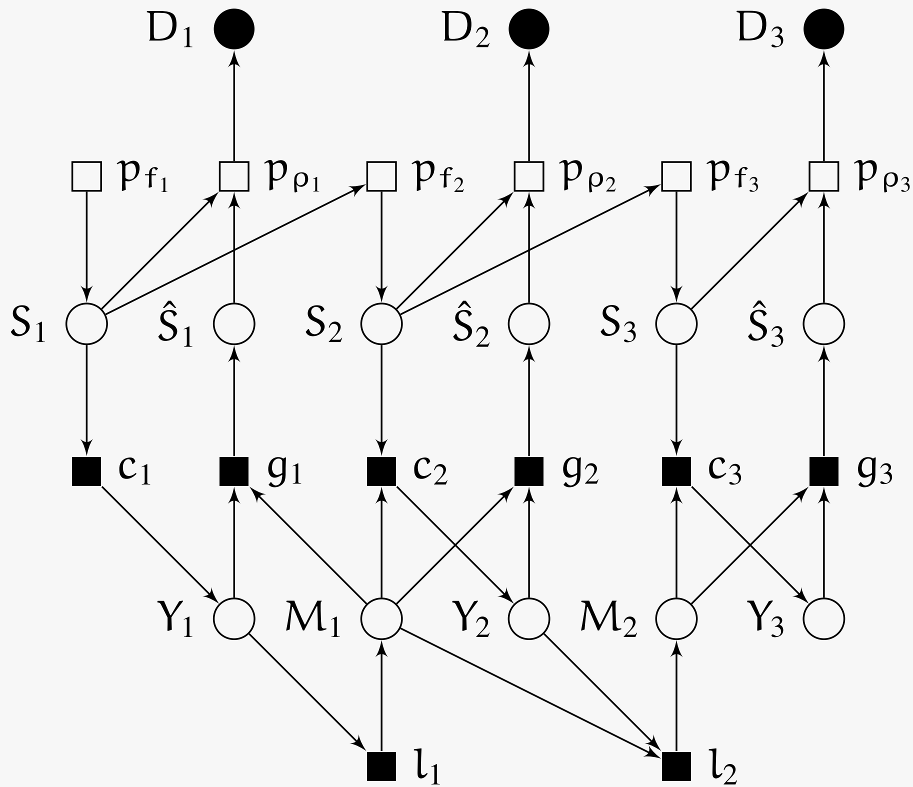
Remove irrelevant nodes



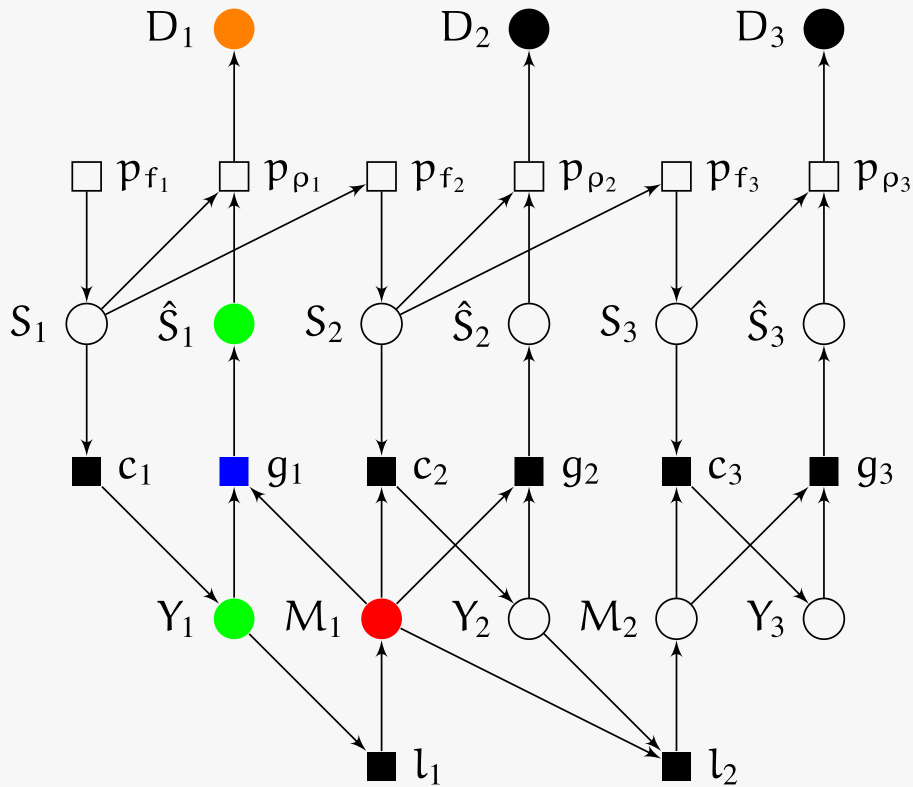
Remove irrelevant nodes



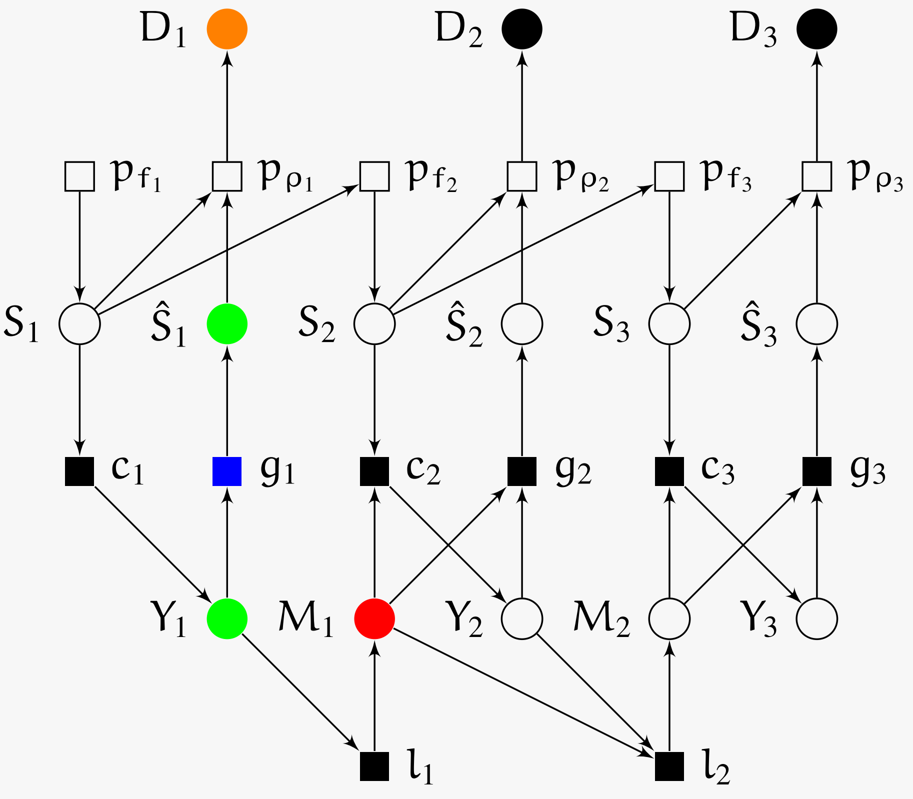
Remove irrelevant nodes



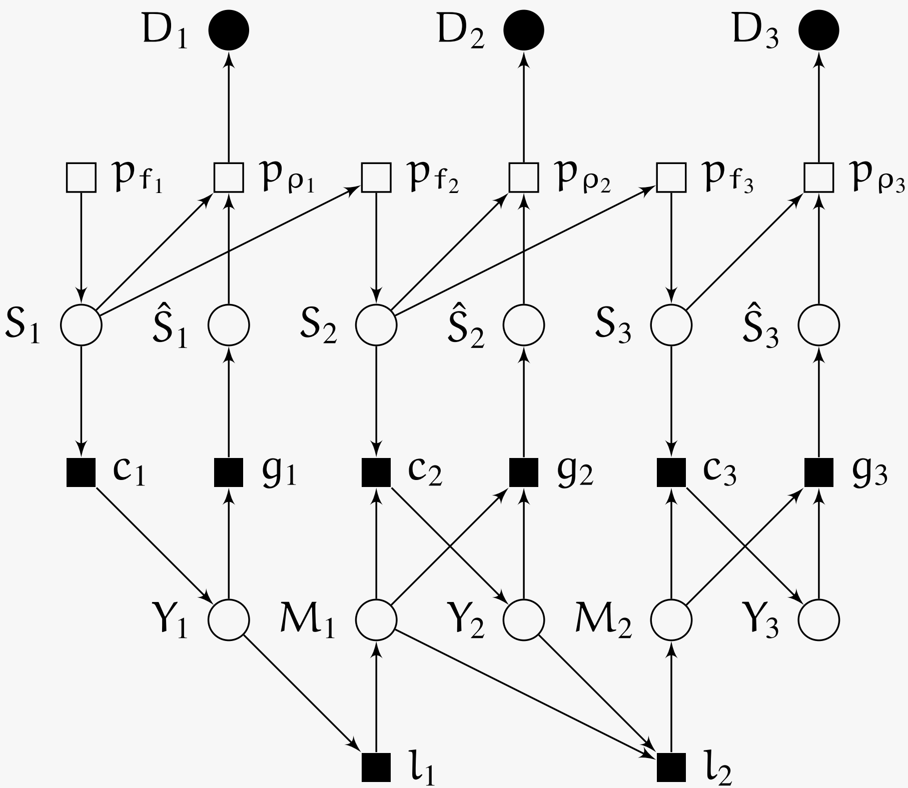
Remove irrelevant nodes



Remove irrelevant nodes



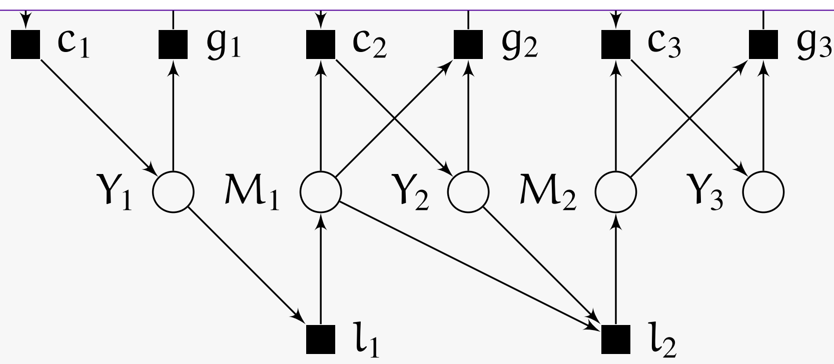
Removing irrelevant nodes



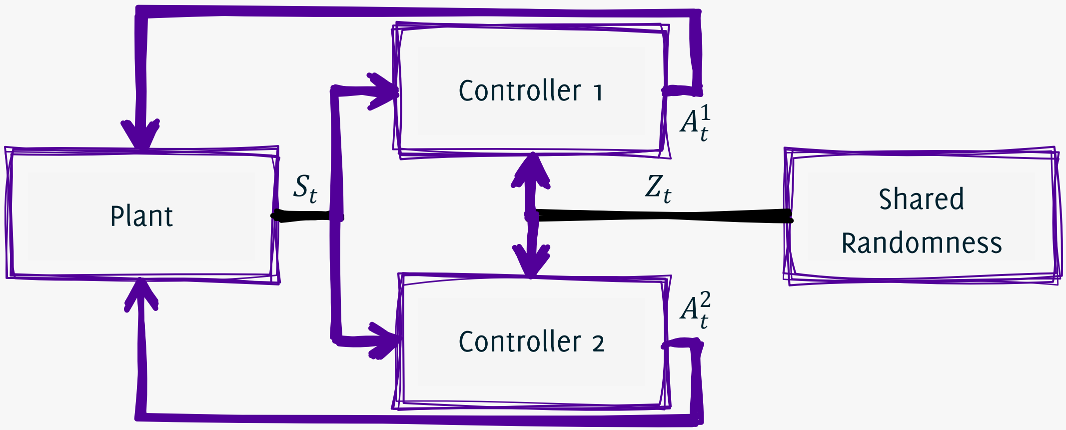
Removing irrelevant nodes

Rederived Witsenhausen's structural result

$$Y_t = c_t(S_t, M_{t-1})$$



Another Example: Shared randomness



Plant: $S_{t+1} = f_t(S_t, A_t^1, A_t^2, W_t)$

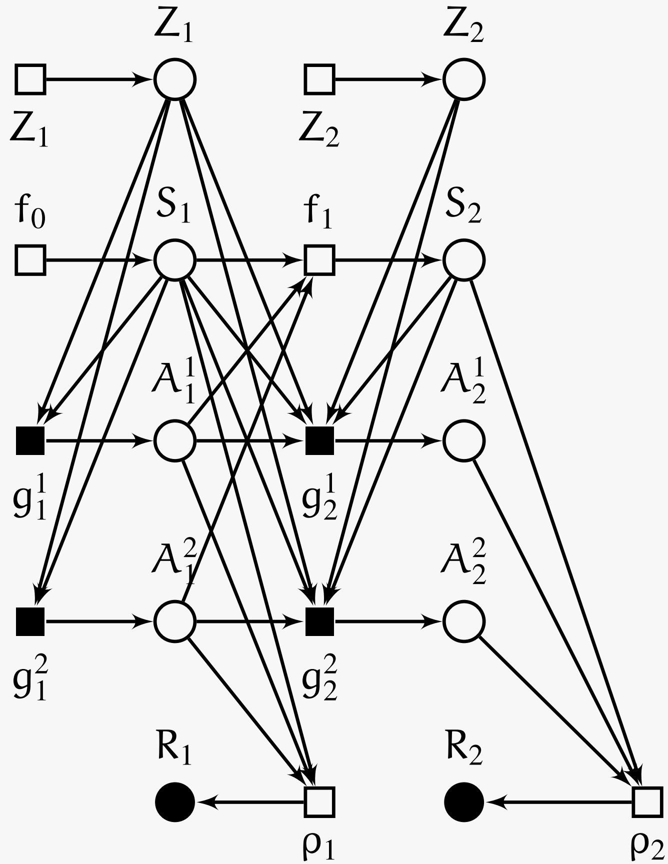
Shared Randomness: $\{Z_t, t = 1, \ldots, T\}$ indep. of rest of system

Control Station 1: $A_t^1 = g_t^1(S^t, A^{1,t-1}, Z^t)$

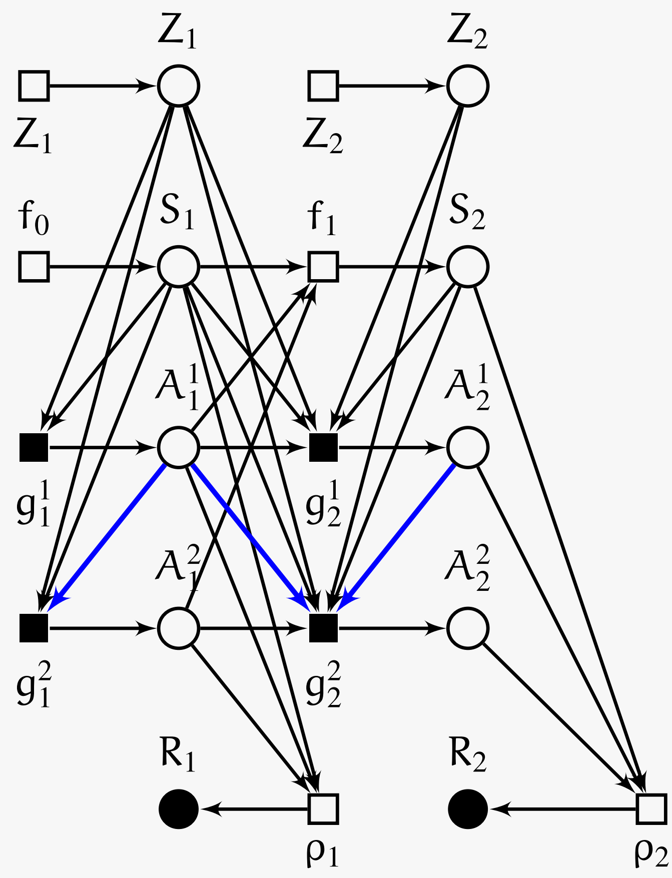
Control Station 2: $A_t^2 = g_t^2(S^t, A^{2,t-1}, Z^t)$

Instantaneous cost: $\rho_t(S_t, A_t^1, A_t^2)$

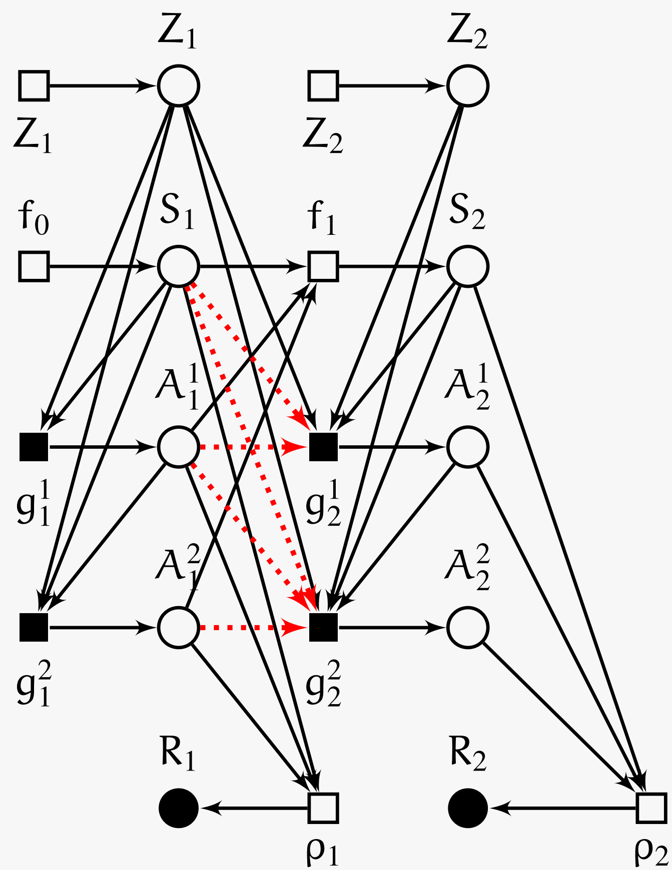
Shared randomness



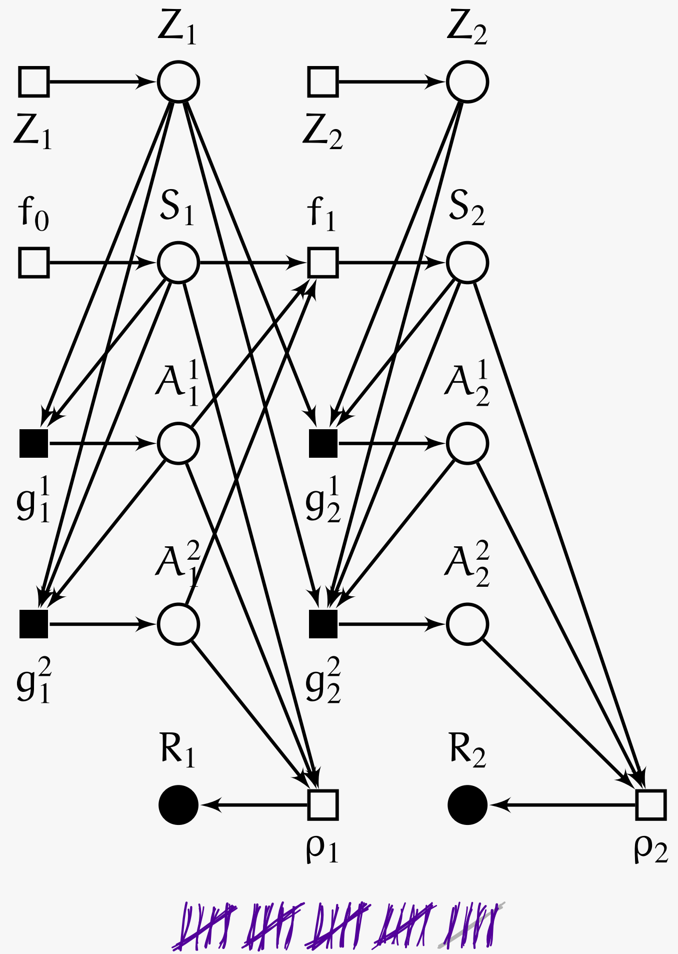
Shared randomness (Step 1)



Shared randomness (Step 2)

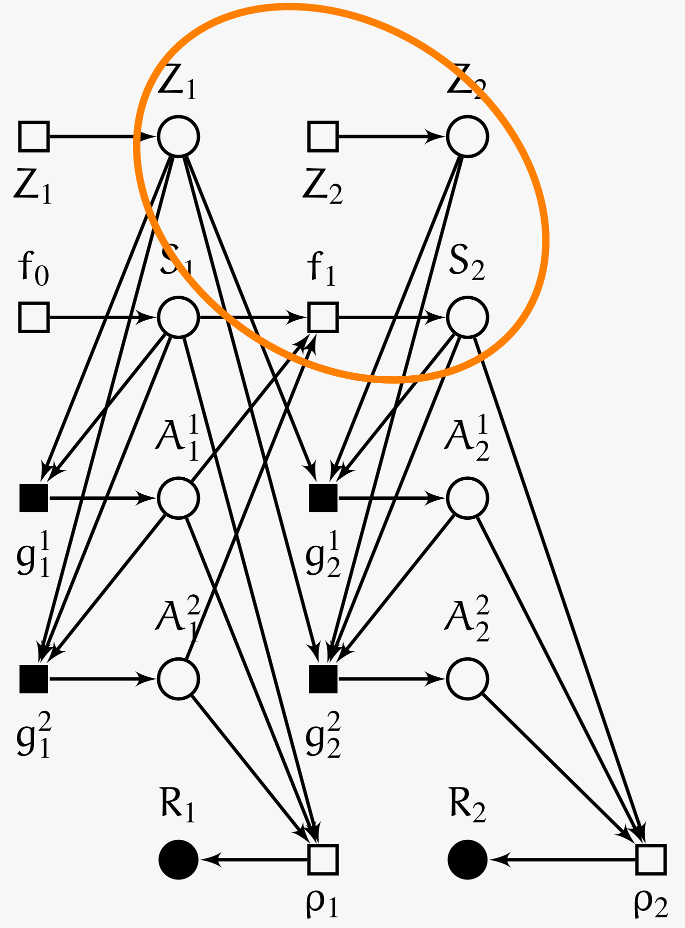


Cannot remove useless sharing

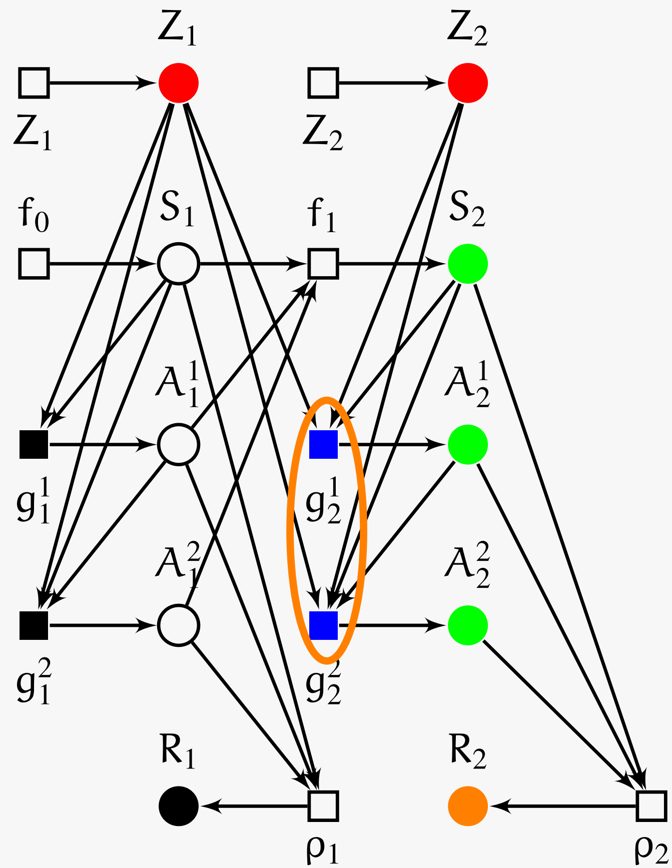


Each agent thinks that the other might use the useless data

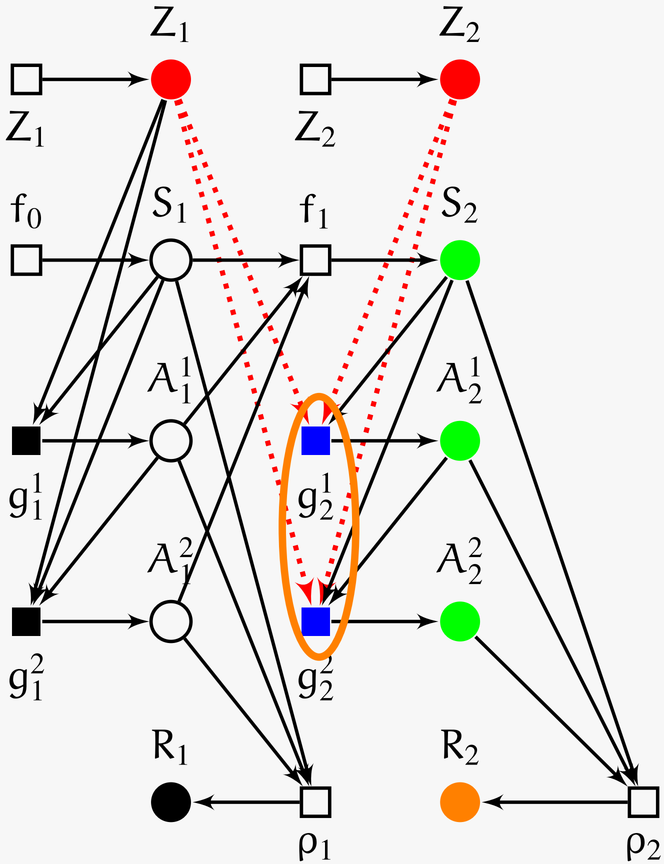
Coordinator's observation



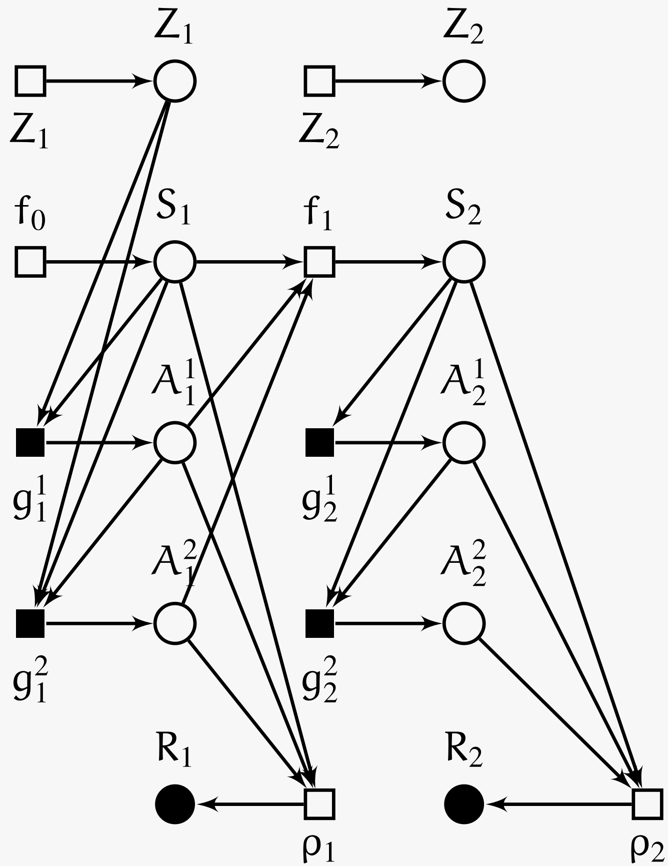
Coordinator



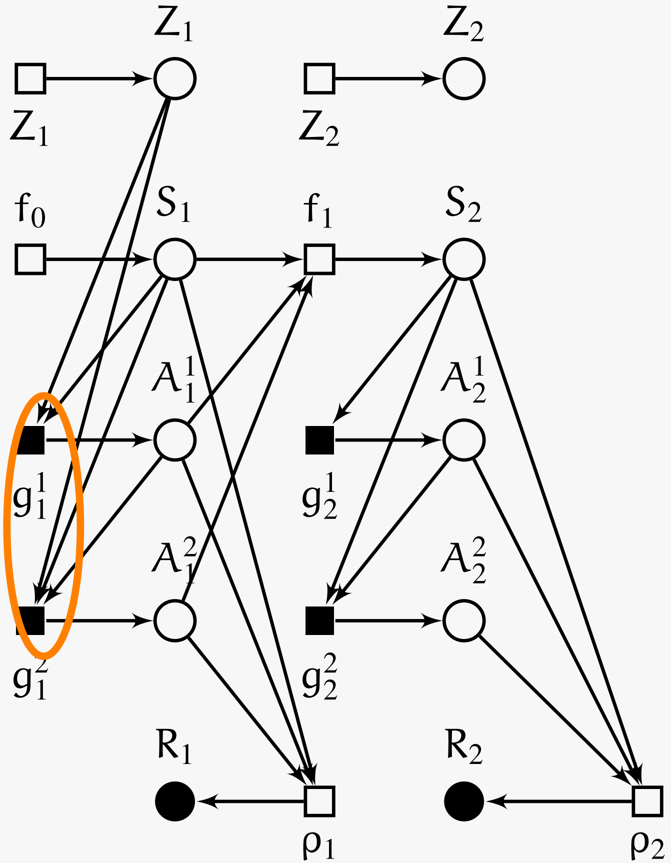
Coordinator



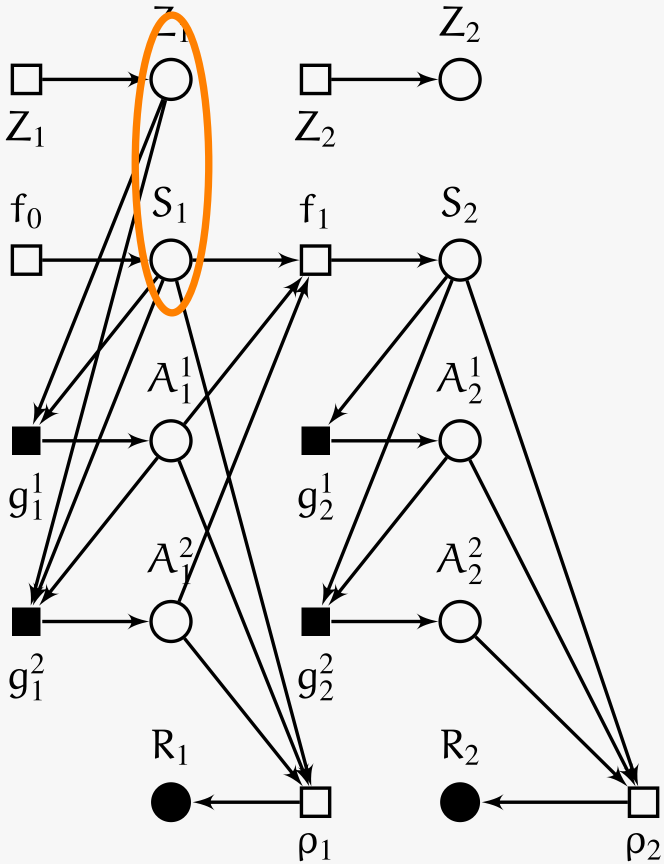
Edges removed



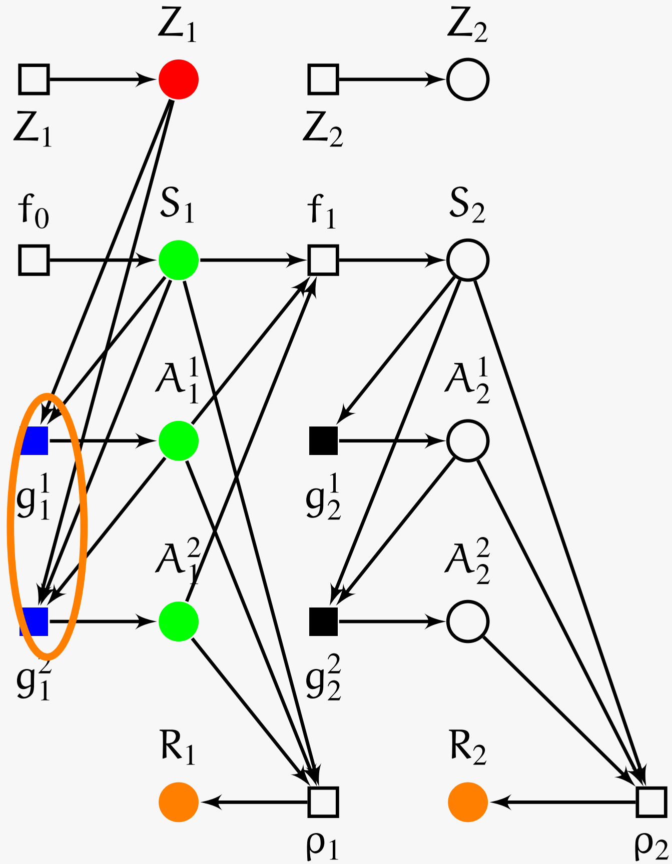
New coordinator



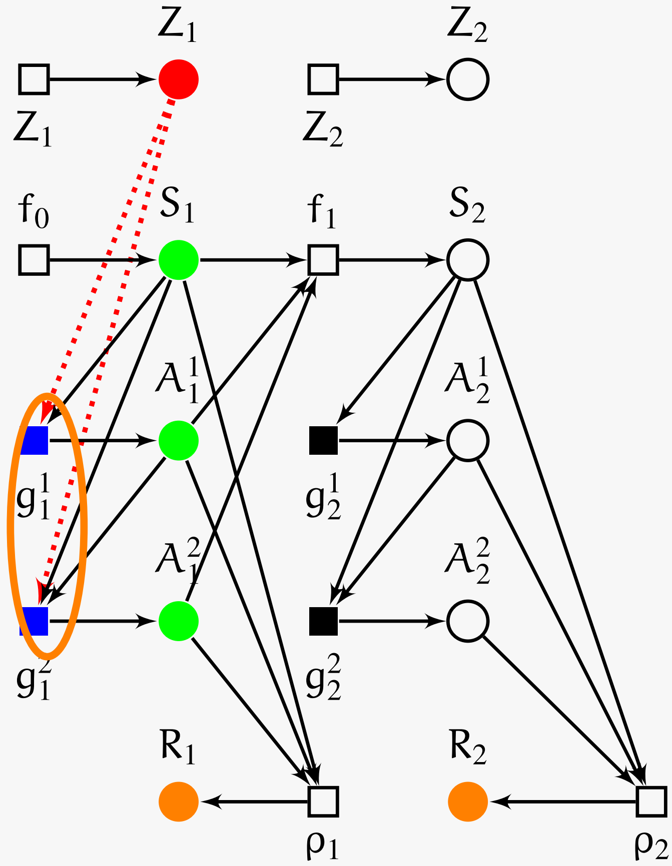
Shared Observation



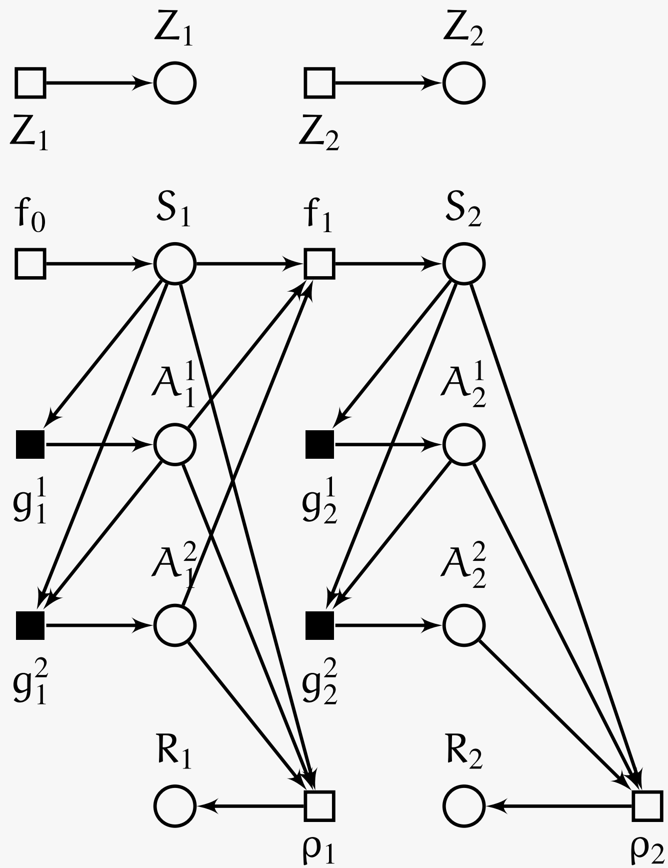
New Coordinator



New Coordinator



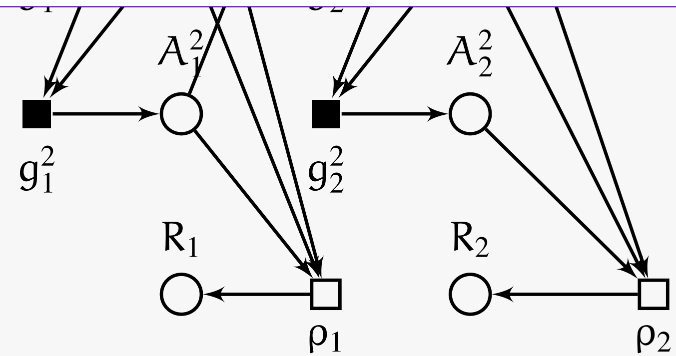
New Coordinator



New Coordinator

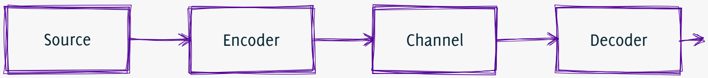
Derived structural result

$$A_t^1 = g_t^1(S_t)$$
$$A_t^2 = g_t^2(S_t, A_t^1)$$



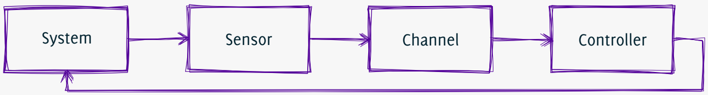
Applications

Real-time communication



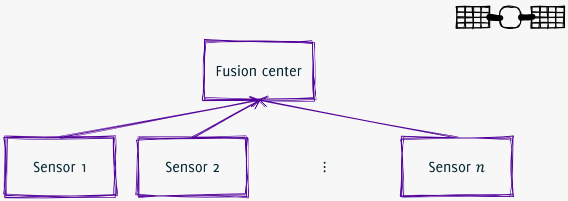
Mahajan-Teneketzis, Trans. IT 09

Control over noisy channels



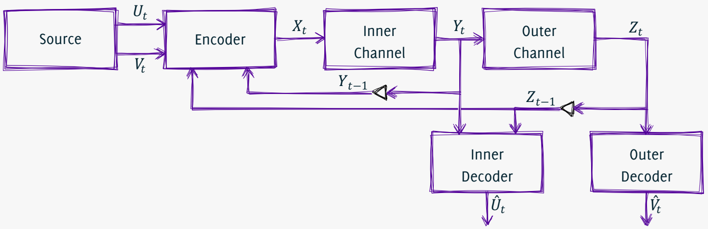
Mahajan-Teneketzis, SICON 09

Sensor scheduling



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

multi-terminal feedback communication



Mahajan, Allerton 09

Conclusion

- Team form for decentralized systems and the notion of equivalence and simplification of team form.
- Modeled as a directed acyclic graph.
  - The simplification process is intuitive
  - The algorithm is efficient and can be automated easily
- Partial results for compressing available data
- Similar idea can be used for sequential decomposition

Thank you

References

- Sequential team form and its simplification using graphical models,

  Mahajan and Tatikonda, Allerton 2009

- Sequential decomposition of sequential teams: applications to real-time communication and networked control systems

  Mahajan, PhD Dissertation, Univ of Michigan, 2008

- Identifying tractable decentralized problems on the basis of information structures

  Mahajan, Nayyar, and Teneketzis, Allerton 2008

- Optimal control strategies in delayed sharing information structures

  Nayyar, Mahajan, and Teneketzis, TAC (submitted 2010)

- Software implementation http://pantheon.yale.edu/~am894/code/teams