Optimal design of sequential teams

Aditya Mahajan

Dept of Electrical Engineering
Yale University

Joint work with:

Ashutosh Nayyar and Demos Teneketzis (UofM); Sekhar Tatikonda (Yale)

Presented at: McGill University, May 20, 2009

Queen's University, May 22, 2009

Outline of the talk

1. Overview of multi-agent systems
   - ▷ Classification : games vs. teams and single-stage vs. multi-stage
   - ▷ Sequential decomposition : what and why
2. Sequential teams with private and common observations
   - ▷ A coordinator of a decentralized system
   - ▷ Sequential decomposition for finite and infinite horizon
3. An example: multi-access broadcast
4. A graphical model for sequential teams
   - ▷ Automated method to find structural results

Multi-agent decentralized systems

- Applications
  - communication networks
  - sensor networks
  - surveillance networks
  - transportation networks
  - control systems
  - monitoring and diagnostic systems
  - multi-robot systems
  - multi-core CPUs
  - . . .

- Salient features
  - System has different components
  - These components know different information
  - The components need to cooperate and coordinate

Question

How do we approach the design of a decentralized multi-agent system?

Classification of multi-agent systems

1. Teams vs. Games

2. Single-stage vs. Multi-stage

   - Sequential vs. non-sequential systems

     - Classical vs. non-classical information structures

> Sequential multi-stage teams with
> non-classical information structures

Sequential Decomposition

- Divide and conquer :

  Exploit sequential and multi-stage nature of the problem

> Convert a one-shot optimal design problem into
> a sequence of nested optimization problems

- Classical information structure (MDP, POMDP)
  - ▷ Dynamic programming

Why consider sequential decomposition

- Finite horizon
  - Brute force search always possible but has high complexity
  - **Provides a systematic way to search for an optimal solution efficiently**
- Infinite horizon
  - Brute force search not possible
  - An arbitrary solution cannot be implemented
  - **Identify qualitative properties to search and implement optimal designs compactly**
- May help in identifying (and proving) other qualitative properties

Sequential Decomposition

- Non-classical information structure

Hans S. Witsenhausen, A standard form for sequential stochastic control, Math. Systems Theory, 7 (1973), pp. 5–11.

Does not work for infinite horizon problems

Aditya Mahajan, Sequential decomposition of sequential teams, Ph.D. Thesis, University of Michigan, Ann Arbor (2008).

Not easy to extend to more than two agents

Sequential decomposition of multi-agent sequential teams that works for finite and infinite horizon

Models considered in this talk

1. multi-agent (>2) teams with private and common observations

   - ▷ **Assume structural property of optimal controller**
   - ▷ Identify a coordinator of the system
   - ▷ Identify an information state sufficient for performance analysis
   - ▷ Optimally control the evolution of information state

   $$\Updownarrow$$

   Markov decision process where

   $$\begin{aligned}\text{State space} &: \text{space of probability measures}\\ \text{Action space} &: \text{space of functions}\end{aligned}$$

2. A graphical model for sequential teams

   - ▷ Automated method to find structural results

Main results

- Multi-agent teams : finite horizon
  - Coordinator's problem is a centralized problem
  - Absence of any common observation $\Rightarrow$ coordinator = designer
- Multi-agent teams : infinite horizon
  - Cannot restrict attention to time invariant designs
  - Can restrict attention to time invariant meta-designs
- Graphical model for sequential teams
  - Structural results can be derived based on conditional independence
  - Conditional independence can be tested efficiently in graphical models

The first model . . .

Teams with common information

- Some models of teams considered in the literature . . .
  - delayed state sharing
  - delayed information sharing
  - delayed observation sharing
  - delayed control sharing
  - delayed belief sharing
  - periodic state sharing
  - periodic information sharing
  - periodic observation sharing
  - periodic control sharing
  - periodic belief sharing

Sharing is good! Can we generalize this?

Teams with common information

- Plant: $x(t+1) = f(x(t), u_{[1:n]}(t), w(t))$
- Observations
  - ▷ Common Observation: $z(t) = o(x(t), u_{[1:n]}(t-1), q(t))$
  - ▷ Private Observation: $y_i(t) = h_i(x(t), n_i(t))$
- Control at agent i: $u_i(t) = g_{i,t}(y_i(1{:}t), u_i(1{:}t-1), z(1{:}t))$
- Design: $G_T := g_{[1:n]}(1{:}T)$ (control laws of all agents for all time)
- Cost
  - ▷ At time t: $c(x(t), u_{[1:n]}(t))$
  - ▷ Of a design G_T: $\mathcal{J}_T(G_T) = E^{G_T}\left\{\sum_{t=1}^{T} c(x(t), u_{[1:n]}(t))\right\}$

Problem P (Finite horizon)

- Given
  - ▷ Alphabets $(\mathcal{X}, \mathcal{Z}, \mathcal{Y}_i, \mathcal{U}_i)$
  - ▷ Plant function f
  - ▷ Observation functions o and h_i
  - ▷ Cost functions c
  - ▷ PMF of $w(t)$, $q(t)$, and $n_i(t)$

- Determine
  - ▷ An optimal design G_T^*

$$\mathcal{J}_T(G_T^*) = J_T^* := \min_{G_T} \mathcal{J}_T(G_T)$$

Salient Features

- Team problem

  All agents have common objective

- Sequential team

  Agent's actions or events in nature do not influence order of agent's actions.

- Data, information and non-classical information structure

$$\sigma(y_i(1{:}t), u_i(1{:}t-1), z(1{:}t)) \nsubseteq \sigma(y_j(1{:}t), u_j(1{:}t-1), z(1{:}t)), \quad i \neq j$$

Solution Approach: Sequential Decomposition

Assumption on controller structure

- Assumption A

> Without loss of optimality, all controllers can compress their past private information $(y_i(1{:}t-1), u_i(1{:}t-1))$ into a sufficient statistic (memory) $m_i(t)$ that takes values in a time invariant space $\mathcal{M}_i$.

- Modified control

$$u_i(t) = \xcancel{g_{i,t}(y_i(1{:}t), u_i(1{:}t-1), z(1{:}t))} = g_{i,t}(y_i(t), m_i(t), z(1{:}t))$$

- Memory update

$$m_i(t+1) = l_{i,t}(y_i(t), m_i(t), z(1{:}t))$$

- Many systems satisfy Assumption A

Solution Approach – An alternate problem

- Problem PC (Problem with a coordinator)

  A centralized system with n components and a coordinator.

- Coordinator's Observations: $z(t)$

- Coordinator's *meta-control*

  $$\gamma_{i,t} = \mathcal{Z}^t \to (\mathcal{Y}_i \times \mathcal{M}_i \to \mathcal{U}_i) \text{ and } \lambda_{i,t} = \mathcal{Z}^t \to (\mathcal{Y}_i \times \mathcal{M}_i \to \mathcal{M}_i)$$

  $$\hat{g}_{i,t} = \gamma_{i,t}(z(1{:}t)) \qquad \hat{l}_{i,t} = \lambda_{i,t}(z(1{:}t))$$

- Component's operation:

  Passively applies the functions supplied by the coordinator.

  $$u_i(t) = \hat{g}_{i,t}(y_i(t), m_i(t)) \qquad m_i(t+1) = \hat{l}_{i,t}(y_i(t), m_i(t))$$

The two problems are equivalent

- $\mathcal{J}_T(\text{Problem PC}) \ge \mathcal{J}_T(\text{Problem P})$
  - Given any strategy (G_T, L_T) of controllers, an equivalent coordinator meta-strategy (Γ_T, Λ_T) can be constructed.

$$\gamma_{i,t}(z(1{:}t)) = g_{i,t}(\cdot,\cdot,z(1{:}t)) \qquad \lambda_{i,t}(z(1{:}t)) = l_{i,t}(\cdot,\cdot,z(1{:}t))$$

- $\mathcal{J}_T(\text{Problem P}) \ge \mathcal{J}_T(\text{Problem PC})$
  - Coordinators actions are measurable at all agents.
  - All agents can implement the coordinator's meta-strategy.

$$g_{i,t}(y_i(t), m_i(t), z(1{:}t)) = \gamma_{i,t}(z(1{:}t))(y_i(t), m_i(t))$$

$$l_{i,t}(y_i(t), m_i(t), z(1{:}t)) = \lambda_{i,t}(z(1{:}t))(y_i(t), m_i(t))$$

The coordinator's problem is centralized!

Information states: Pr(state | information)

- State (for coordinator)

$$s_1(t) = (x(t), y_{[1:n]}(t), u_{[1:n]}(t-1), m_{[1:n]}(t-1))$$

$$s_i(t) = (x(t), y_{[1:k]}(t), u_{[1:i-1]}(t), \quad m_{[1:i-1]}(t), m_{[i:n]}(t-1)), \quad i = 2, \ldots, n$$

Cost: $\quad c(x(t), u_{[1:n]}(t)) = \hat{c}(s_n(t))$

- Information (at the coordinator)

$$\sigma(z(1{:}t); \underbrace{\hat{g}_{[1:n]}(1{:}t-1), \hat{l}_{[1:n]}(1{:}t-1), \hat{g}_{[1:i-1]}(t), \hat{l}_{[1:i-1]}(t)}_{\varphi_i(t)}, \cancel{\gamma_{[1:n]}(1{:}t-1)}, \cancel{\lambda_{[1:n]}(1{:}t-1)})$$

- Information state

$$\pi_i(t)(z(1{:}t), \varphi_i(t)) = \Pr\left(S_i(t) \mid \sigma(z(1{:}t); \varphi(t))\right) = \Pr^{\varphi_i(t)}\left(S_i(t) \mid Z(1{:}t) = z(1{:}t)\right)$$

Properties of information states

- Update

  There exists functions F_i, $i = 1, \ldots, n$, such that

$$\pi_1(t) = F_1(\pi_n(t-1), \hat{g}_k(t-1), \hat{l}_k(t-1))$$

$$\pi_i(t) = F_1(\pi_{i-1}(t), \hat{g}_{i-1}(t), \hat{l}_{i-1}(t)), \qquad i = 2, \ldots, n$$

- Information state update is independent of the meta-strategy
- Information state is measurable at all controllers
- Information state is “time-invariant”

Sequential Decomposition

- Initialization

$$V_{n,T}(\tilde{\pi}_n) = E\{\hat{c}(S_n \,|\, \pi_n(T) = \tilde{\pi}_n)\}$$

- Backward recursion
  - For $i = 1, \ldots, n-1$, and $t = 1, \ldots, T$,

$$V_{i,t}(\tilde{\pi}_i) = \inf_{\hat{g}_{i,t}, \hat{l}_{i,t}} \left[E^{\hat{g}_{i,t}, \hat{l}_{i,t}} \{V_{i+1,t}(\pi_{i+1}(t)) \,|\, \pi_i(t) = \tilde{\pi}_i\} \right]$$

  - For $t = 1, \ldots, T$,

$$V_{n,t}(\tilde{\pi}_n) = E\{\hat{c}(S_n \,|\, \pi_n(t) = \tilde{\pi}_n)\} + \inf_{\hat{g}_{n,t}, \hat{l}_{n,t}} \left[E^{\hat{g}_{n,t}, \hat{l}_{n,t}} \{V_{1,t+1}(\pi_1(t+1)) \,|\, \pi_n(t) = \tilde{\pi}_n\} \right]$$

An interpretation of the solution

- All the agents decide what to do before the system starts operating

In the presence of common information, the coordinator
allows the agents to adapt to the component
of the same path that is commonly observed

In the absence of common information, the
coordinator is same as the system designer

- Information states store and process the common information and past policies more efficiently.

Infinite horizon problem

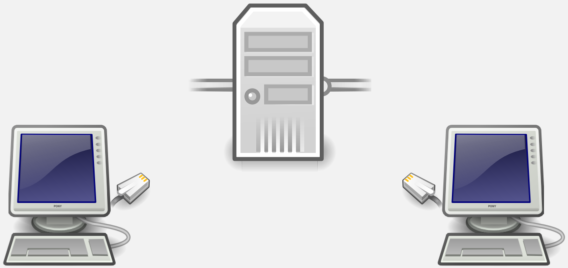
- Information states belong to "time invariant" spaces, so the sequential decomposition extends to infinite horizon.
- Meta-strategy is stationary $\Rightarrow$ meta control laws are time invariant.
- Actual control laws change with time
- The coordinator's viewpoint makes it easy to find and implement optimal time varying control laws.

Summary so far ...

- Started with a general n agent sequential team with common observations.
- Assume certain structural results exist (private information can be compressed to a sufficient statistic)
- Consider an alternate problem (with passive agents and a coordinator)
- Both problems are equivalent
- The coordinator's problem is centralized!
- Coordinator chooses meta control laws (control laws that choose control laws)
- Each step of the sequential decomposition is a functional problem

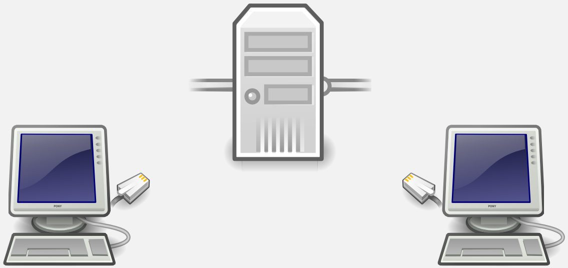
An example

Multiaccess broadcast



Multiaccess broadcast

- MAB Channel
  - Single user transmits $\Rightarrow$ 
  - Both users transmit $\Rightarrow$ 



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

Multiaccess broadcast

- Channel Feedback

  Both users know if there was no Tx, successful Tx, or collision

- Policy of user

$$u_i(t) = g_{i,t}(x_i(1{:}t), u_i(1{:}t-1), z(1{:}t))$$

- Objective : Maximize throughput (or minimize delay)
  - Avoid collisions
  - Avoid idle

History of multiaccess broadcast

- Hluchyj and Gallager, Multiaccess of a slotted channel by finitely many users, NTC 81.
  - ▷ Considered symmetric arrival rates
  - ▷ Restricted attention to “window protocols”
- Ooi and Wornell, Decentralized control of multiple access broadcast channels, CDC 96.
  - ▷ Considered a relaxation of the problem
  - ▷ Numerically find optimal performance of the relaxed problem
  - ▷ Hluchyj and Gallager’s scheme meets this upper bound
- AI Literature
  - ▷ Consider the case of asymmetric arrival rates
  - ▷ Approximate heuristic solutions for small horizons

Optimal Solution

- Structural Result

$$u_i(t) = g_{i,t}(x_i(t), \cancel{x_i(1{:}t-1)}, \cancel{u_i(1{:}t-1)}, z(1{:}t))$$

- Assumption A is satisfied $\Rightarrow$ sequential decomposition

- Further simplifications

  - Information states

$$P((x_1(t), x_2(t)) \mid z(1{:}t)) \equiv (P(x_1(t) \mid z(1{:}t)), P(x_2(t) \mid z(1{:}t)))$$

  - Actions: $\hat{g}_{i,t}(0) = 0,\ \hat{g}_{i,t}(1) {=} ??$

Optimal Solution

- Sequential decomposition same as that of a POMDP with finite state and action space.

- 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!

Back to general teams
Structural properties?

A model for sequential team

- Components of a sequential team
  - ▷ A set N of indices of system variables $\{X_n, n \in N\}$ and a partial order $\prec$ on N
    - – $A \subset N$, variables generated by DM
    - – $N \setminus A$, variables generated by nature
    - – $R \subset N$, reward variables
  - ▷ Finite sets $\{\mathcal{X}_n,\ n \in N\}$ of state spaces of X_n
  - ▷ $\{I_n,\ n \in N\}$, such that for all $i \in I_n$, $i \prec n$. $\mathcal{I}_n = \prod_{i \in I_n} \mathcal{X}_i$
  - ▷ $F_{N \setminus A} = \{f_n,\ n \in N \setminus A\}$, where f_n is a stochastic kernel from $\mathcal{I}_n$ to $\mathcal{X}_n$.
- Design
  - ▷ $G_A = \{g_n,\ n \in A\}$, where g_n is a decision rule from $\mathcal{I}_n$ to $\mathcal{X}_n$

A model for sequential team

- Probability measure induced by a design

$$P^{G_A}(X_N) = \prod_{n \in N \setminus A} f_n(X_n | I_n) \prod_{n \in A} I\left[X_n = g_n(I_n)\right]$$

- Optimization problem

Minimize $E\left\{\sum_{n \in R} X_n\right\}$, where the expectation is with respect to P^{G_A}.

Generality of the Model

- Witsenhausen's intrinsic model

Hans S. Witsenhausen, On information structures, feedback and causality, SIAM Journal of Control, 9 (1971), pp. 149-160.

partial order $\Leftrightarrow$ sequentiality

- Witsenhausen's sequential control model

Hans S. Witsenhausen, A standard form for sequential stochastic control, Math. Systems Theory, 7 (1973), pp. 5–11.

- Witsenhausen's equivalent control model

Hans S. Witsenhausen, Equivalent stochastic control problems, Math. Controls, Signals and Systems, 1 (1988), pp. 3–11.

A graphical model for sequential teams

- Directed Acyclic Factor Graph

- Factor Graph

  - Variable node $n \equiv$ system variable X_n
  - Factor node $\tilde{n} \equiv$ stochastic kernel f_n or decision rule g_n

- Directed Graph

  - $(i,\tilde{n})$, for each $n \in N$ and $i \in I_n$
  - $(\tilde{n}, n)$, for each $n \in N$

- Acyclic Graph

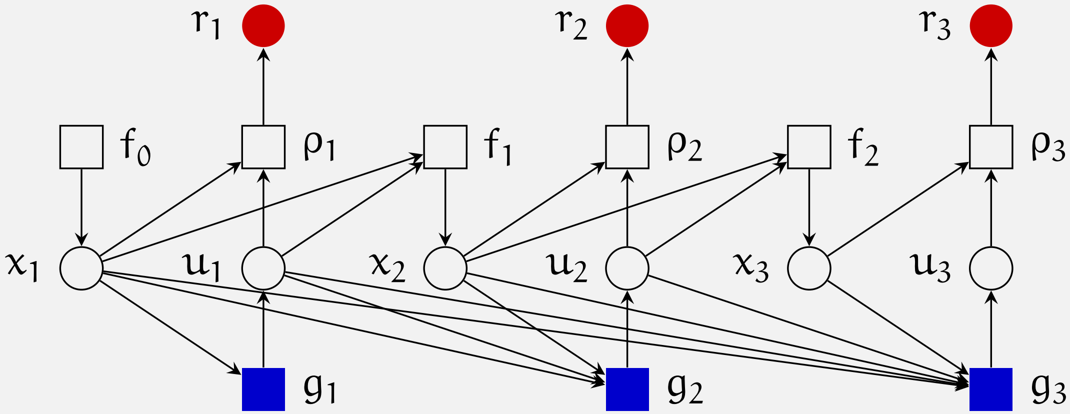
  - Partial order on N implies acyclic graph

An example: MDP

- Mathematical Model
  - ▷ Plant: $f_{t-1}(x_t | x_{t-1}, u_{t-1})$
  - ▷ Control: $u_t = g_t(x_1, \ldots, x_t, u_1, \ldots, u_{t-1})$
  - ▷ Reward: $r_t = \rho_t(x_t, u_t)$

An example: MDP

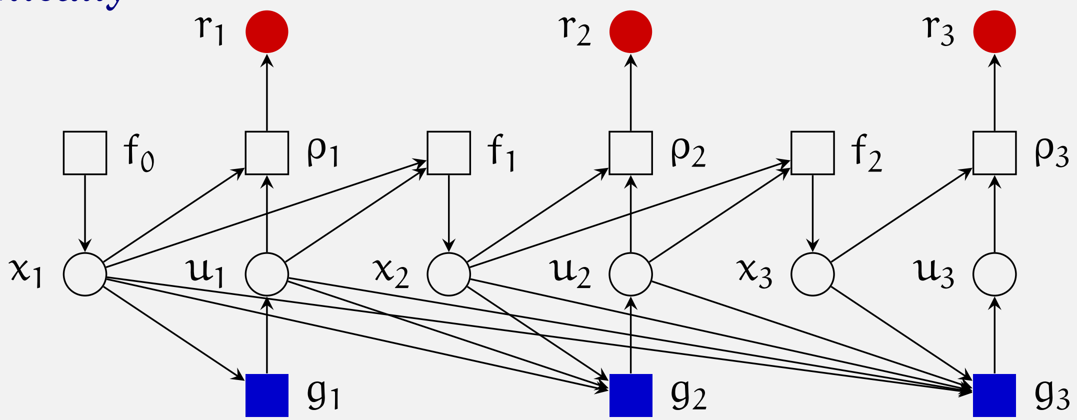
- Mathematical Model
  - Plant: $f_{t-1}(x_t | x_{t-1}, u_{t-1})$
  - Control: $u_t = g_t(x_1, \ldots, x_t, u_1, \ldots, u_{t-1})$
  - Reward: $r_t = \rho_t(x_t, u_t)$
- Graphical Model



Structural Results for MDP

Without loss of optimality, $u_t = g_t(x_t)$

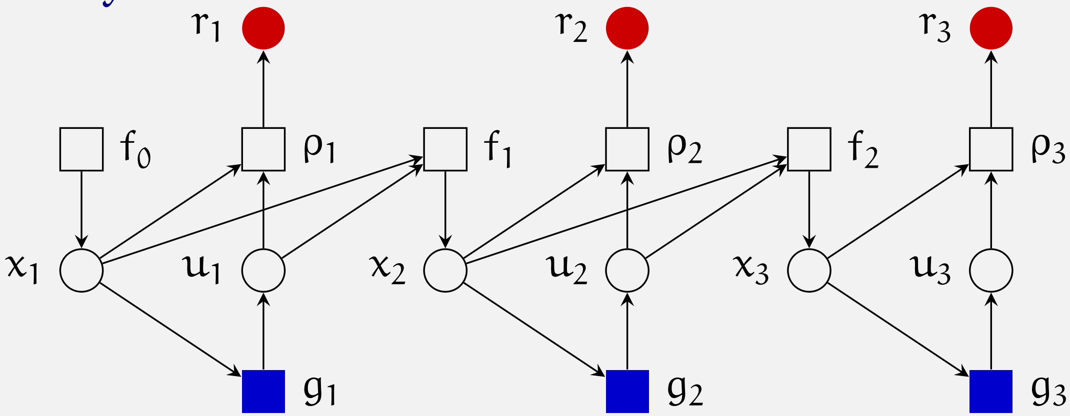
- Graphically



Structural Results for MDP

Without loss of optimality, $u_t = g_t(x_t)$

- Graphically



Structural results

- The main idea

If some data available at a DM is independent of future rewards given the control action and other data at the DM, then that data can be ignored

Can we automate this process?

Struct. result $\equiv$ cond. independence

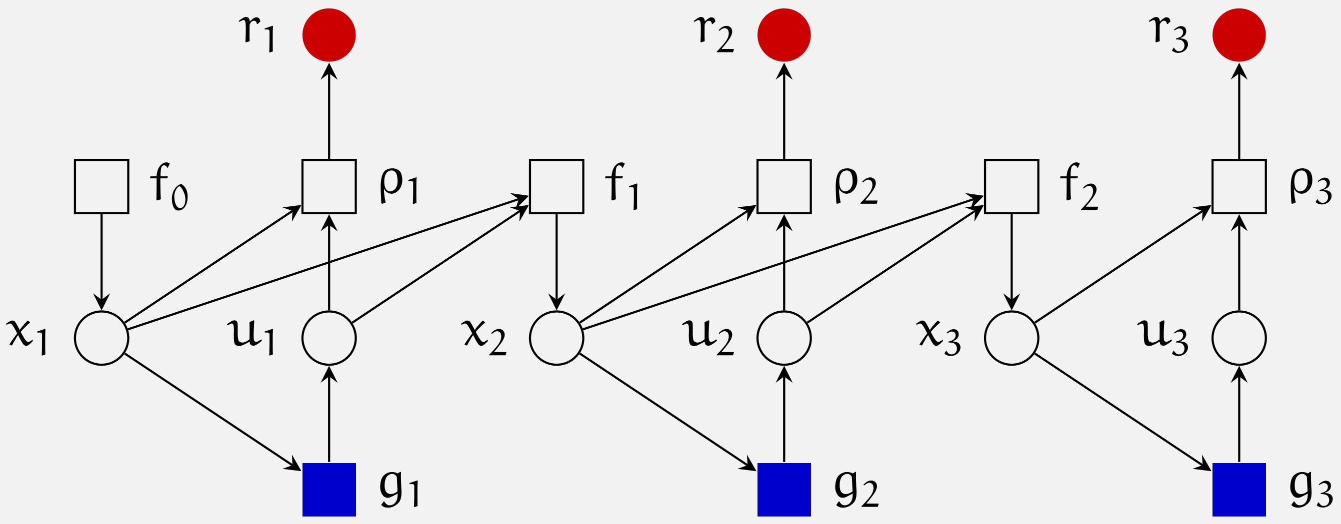
Graphical models can easily
test conditional independence

Graphical models

- Terminology
  - parents(n): All nodes m such that $m \quad n$
  - childred(n): All nodes m such that $n \quad m$
  - ancestors(n): All nodes m such that there is a directed path from m to n
  - descendants(n): All nodes m such that there is a directed path from n to m

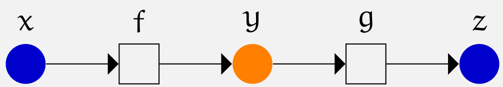
- In terms of teams
  - Parents of a control (factor) node = data observed by controller
  - Children of a control node = control action
  - Ancestors of a control node = all nodes that affect the data observed
  - Descendants of a control node = all nodes that are affected by the control action

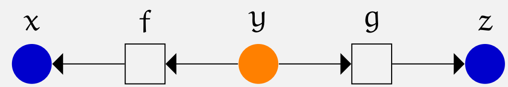
Graphical Models

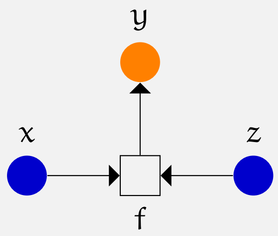


Conditional independence

- Three canonical graphs to verify $x \perp\!\!\!\perp z \mid y$







 Markov chain

 Hidden cause

 Explanation

- Blocking of a trail

  A trail from a to b is blocked by C if $\exists$ a node v on the trail such that either:

  - either $\rightarrow v \rightarrow$, $\leftarrow v \leftarrow$, or $\leftarrow v \rightarrow$, and $v \in C$
  - $\rightarrow v \leftarrow$ and neither v nor any of v's descendants are in S.

Conditional independence

- d-separation

  A is d-separated from B by S if all trails from A to B are blocked by S

- Conditional independence

  For any probability measure P that factorizes according to a DAFG,

$$A \text{ d-separated from } B \text{ by } C \text{ implies}$$
$$X_A \text{ is conditionally independent of } X_B \text{ given } X_C,\ P \text{ a.s.}$$

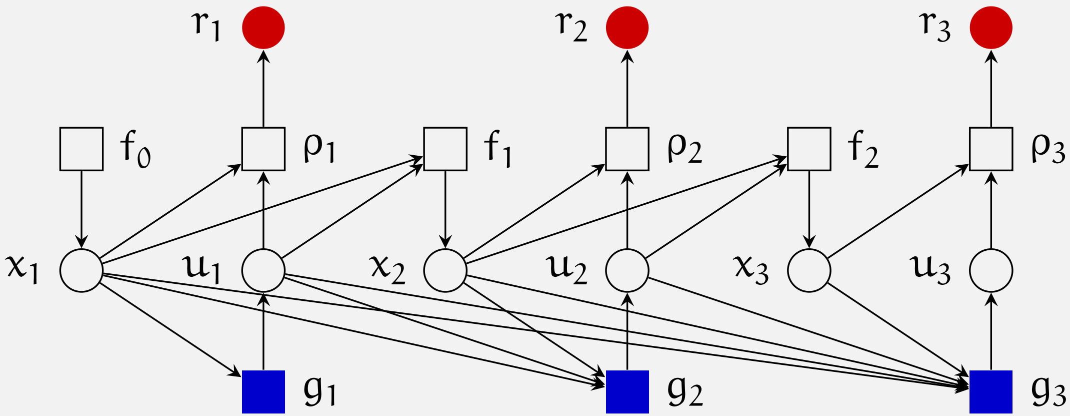
- Efficient algorithms to verify d-separation

  ▷ Moral graph ▷ Bayes Ball

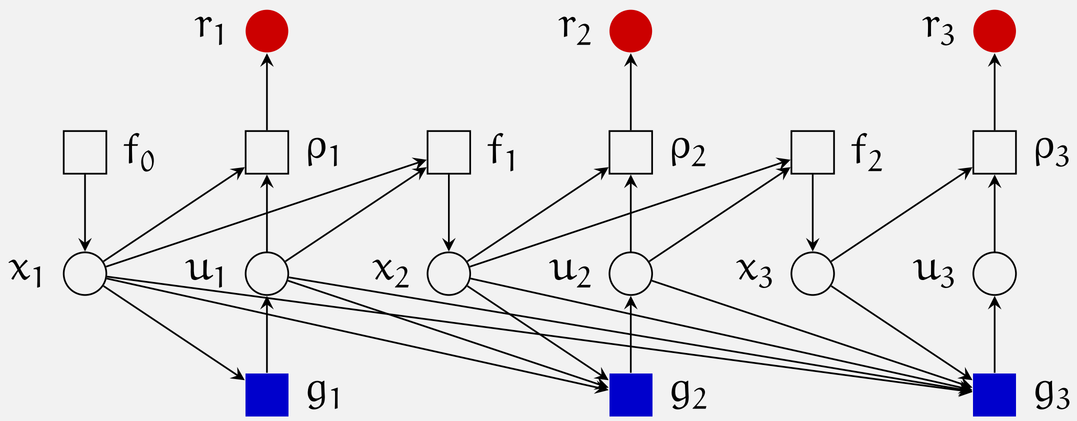
Automated Structural results

- First attempt
  - Dependent rewards: $R_d(\tilde{n}) = R \cap \text{descendants}(\tilde{n})$
  - Irrelevant data: At a control node $\tilde{n}$, and parent i is irrelevant if $R_d(\tilde{n})$ is d-separate from i given $\text{parents}(\tilde{n}) \cup \text{childred}(\tilde{n}) \setminus \{i\}$
  - Requisite data: All parents that are not irrelevant
- Structural result
  - Without loss of optimality, we can choose $u_n = g_n(\text{requisite}(\tilde{n}))$

Structural Results for MDP – Step 1

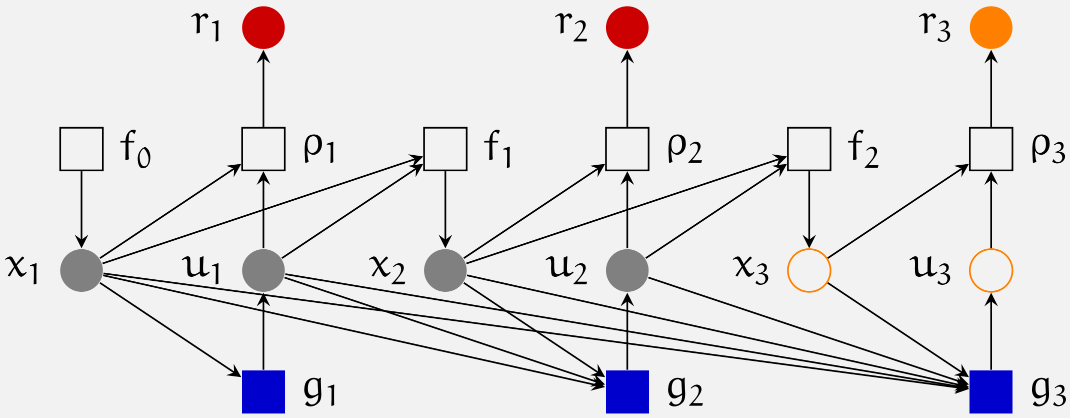


Structural Results for MDP – Step 1



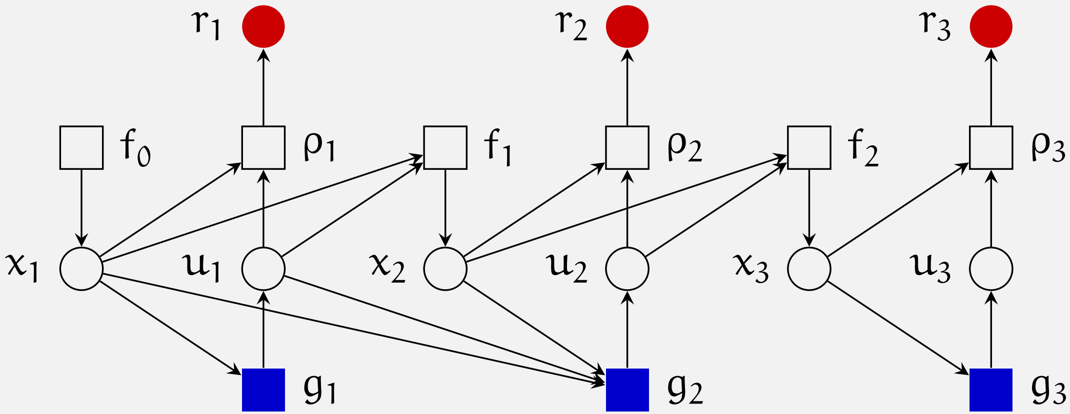
- Pick node g_3.
  - ▷ Original $u_3 = g_3(x_1, x_2, x_3, u_1, u_2)$
  - ▷ requisite(g_3) = $\{x_3\}$
  - ▷ Thus, $u_3 = g_3(x_3)$

Structural Results for MDP – Step 1

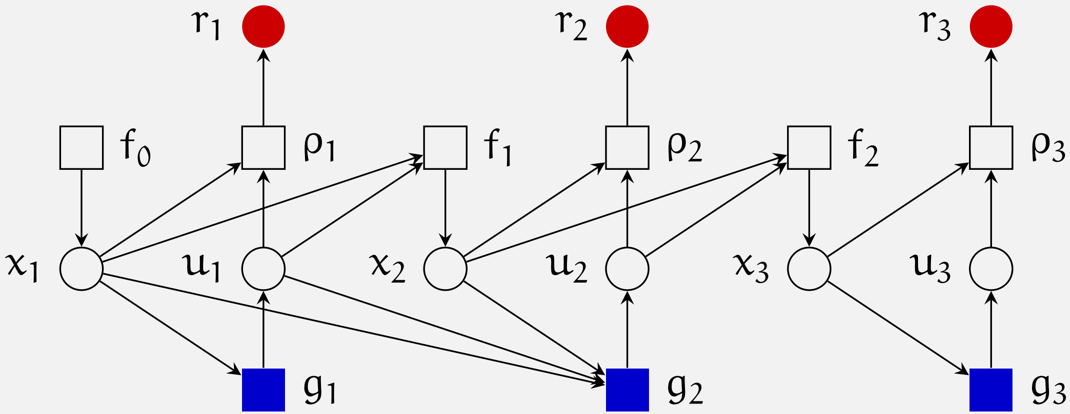


- Pick node g_3.
  - ▷ Original $u_3 = g_3(x_1, x_2, x_3, u_1, u_2)$
  - ▷ requisite(g_3) = $\{x_3\}$
  - ▷ Thus, $u_3 = g_3(x_3)$

Structural Results for MDP – Step 2

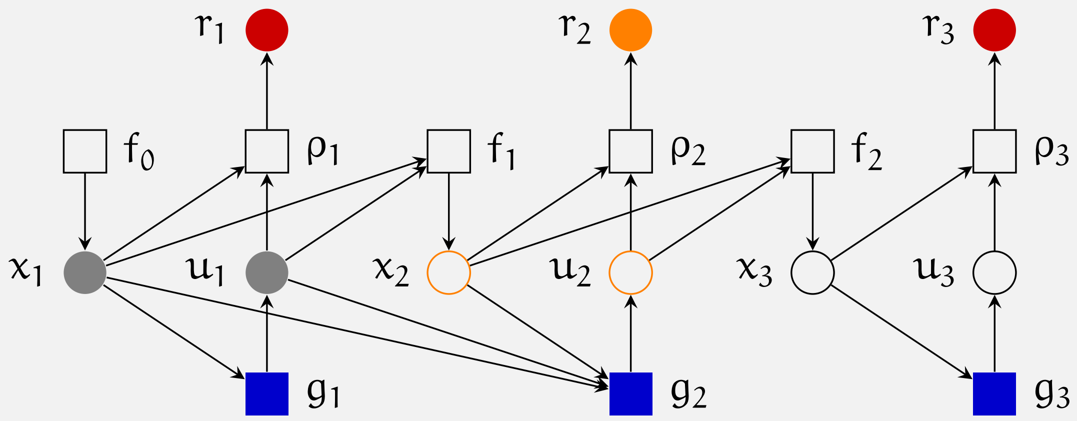


Structural Results for MDP – Step 2



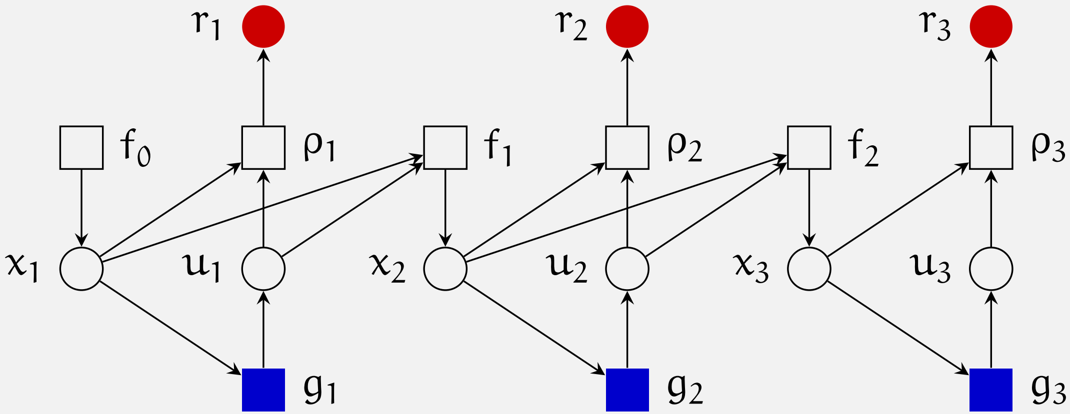
- Pick node g_2.
  - ▷ Original $u_2 = g_2(x_1, x_2, u_1)$
  - ▷ requisite(g_2) = $\{x_2\}$
  - ▷ Thus, $u_2 = g_2(x_2)$

Structural Results for MDP – Step 2



- Pick node g_2.
  - ▷ Original $u_2 = g_2(x_1, x_2, u_1)$
  - ▷ requisite(g_2) = $\{x_2\}$
  - ▷ Thus, $u_2 = g_2(x_2)$

Structural Results for MDP – Step 2



Almost there . . .

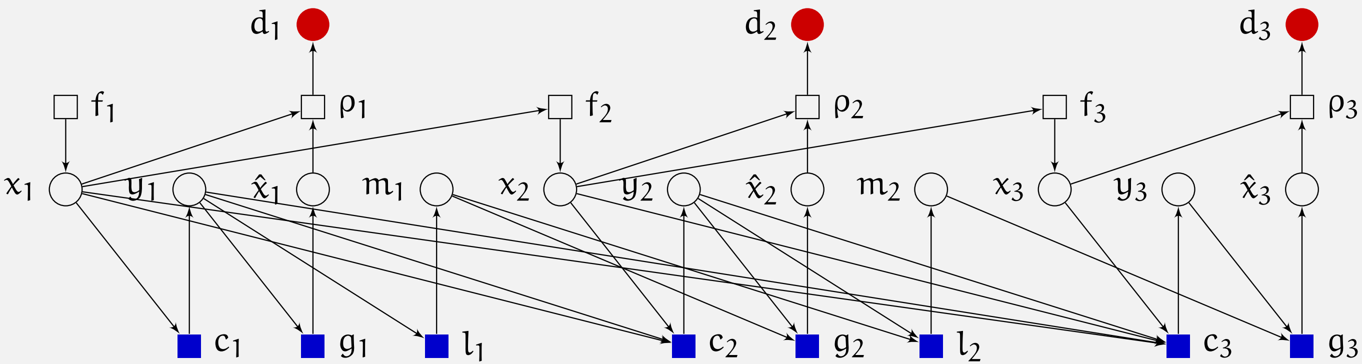
A real-time source coding problem

Hans S. Witsenhausen, On the structure of real-time source coders,
Bell Systems Technical Journal, vol 58, no 6, pp 1437-1451, July-August 1979

- Mathematical Model

  - Source: First order Markov $f_t(x_{t+1}|x_t)$
  - Real-time source coder: $y_t = c_t(x(1{:}t), y(1{:}t-1))$
  - Finite memory decoder: $\hat{x}_t = g_t(y_t, m_{t-1})$
  - $m_{t+1} = l_t(y_t, m_{t-1})$
  - Cost: $\rho(d_t|x_t, \hat{x}_t)$

Graphical model for real-time communication



Need to take care of deterministic functions!

Functionally determined nodes

- Functionally determined
  - ▷ X_B is functionally determined by X_A if $X_B \perp\!\!\!\perp X_N \mid X_A$
- Conditional independence with functionally determined nodes
  - ▷ Can be checked using D-separation
  - ▷ Similar to d-sep: in the defn of blocking change "in C" by "is func detm by C"
- Blocking of a trail (version that takes care of detm nodes)

A trail from a to b is blocked by C if $\exists$ a node v on the trail such that either:

- either $\rightarrow v \rightarrow$, $\leftarrow v \leftarrow$, or $\leftarrow v \rightarrow$, and v is functionally determined by C
- $\rightarrow v \leftarrow$ and neither v nor any of v's descendants are in S.

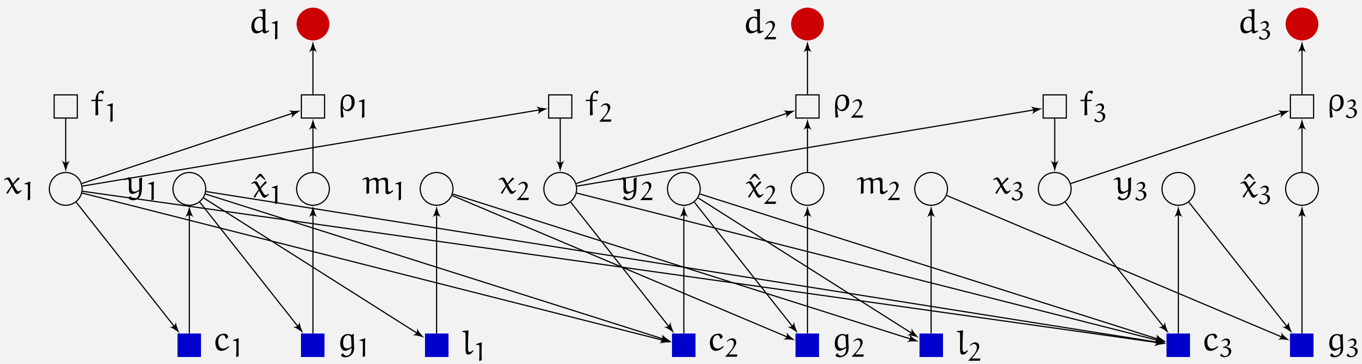
Automated Structural results

- Second attempt
  - Irrelevant data: Change d-separation by D-separation
  - Requisite data: All parents that are not irrelevant
- Structural result
  - Without loss of optimality, we can choose

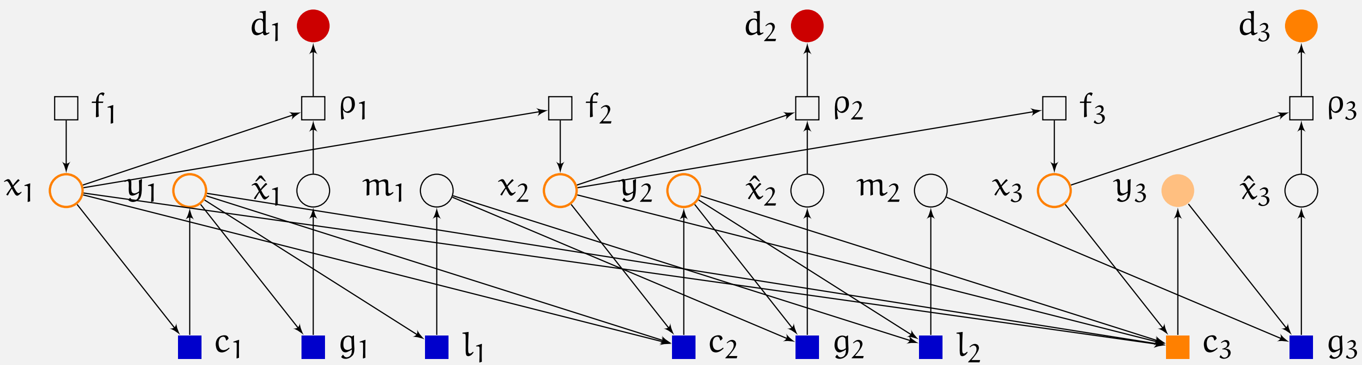
$$u_n = g_n(\text{requisite}(\tilde{n}), \text{functionally\_detm}(\tilde{n}) \cap \text{ancestors}(R_d(\tilde{n})))$$

Proof: use policy independence of conditional expectation and follow the steps of the three step lemma.

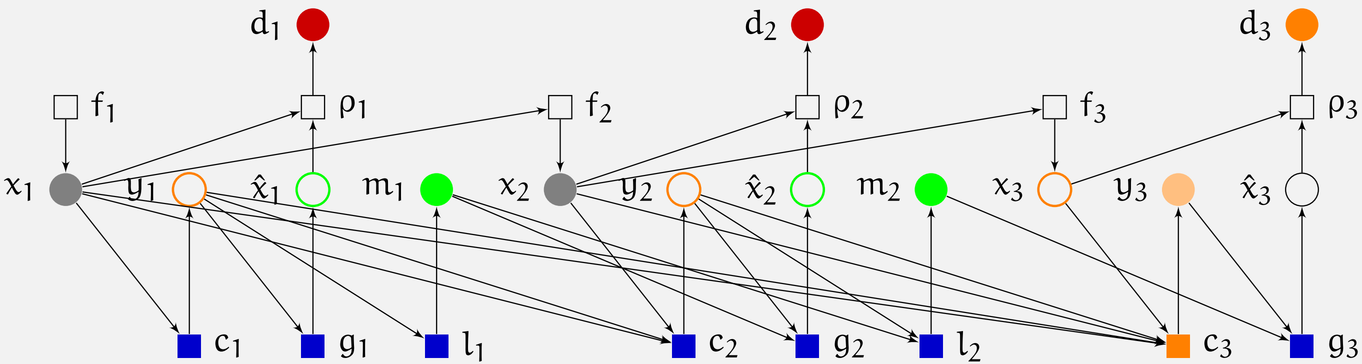
Lets try this!



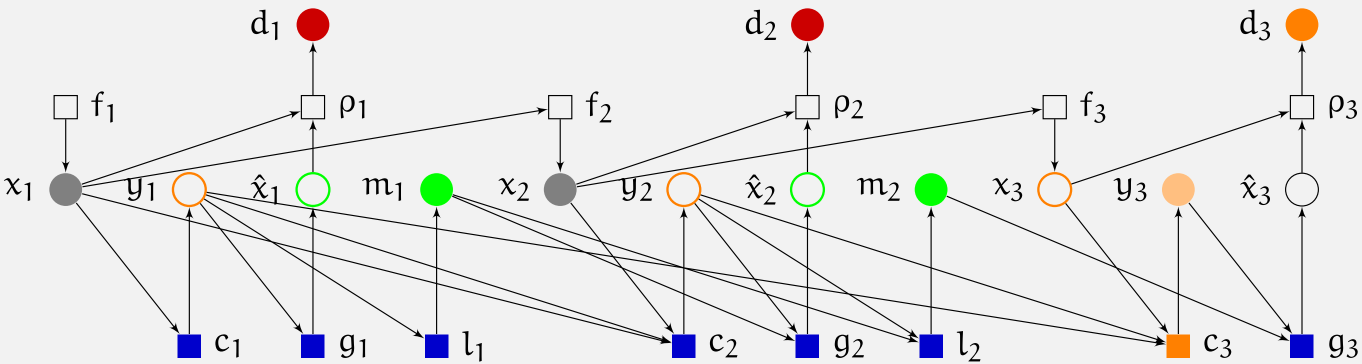
Structural Results for Dec MDP – Step 1



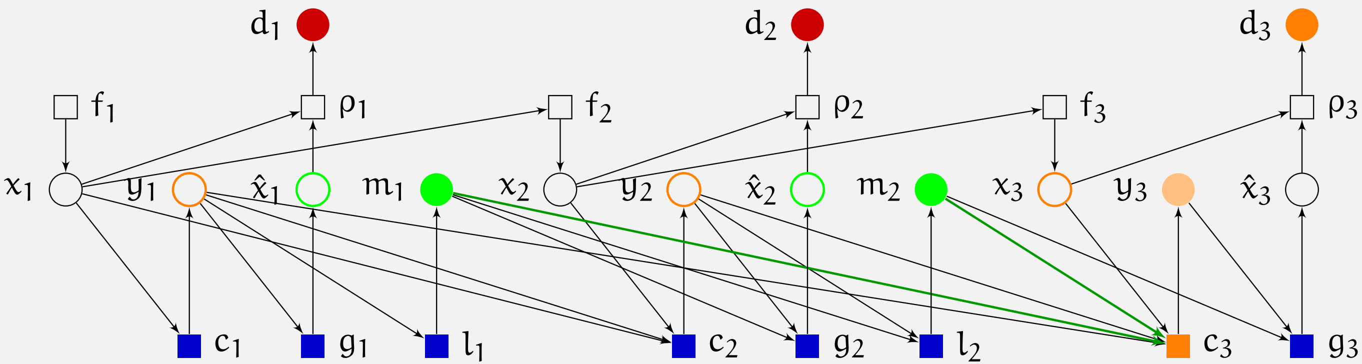
Structural Results for Dec MDP – Step 2



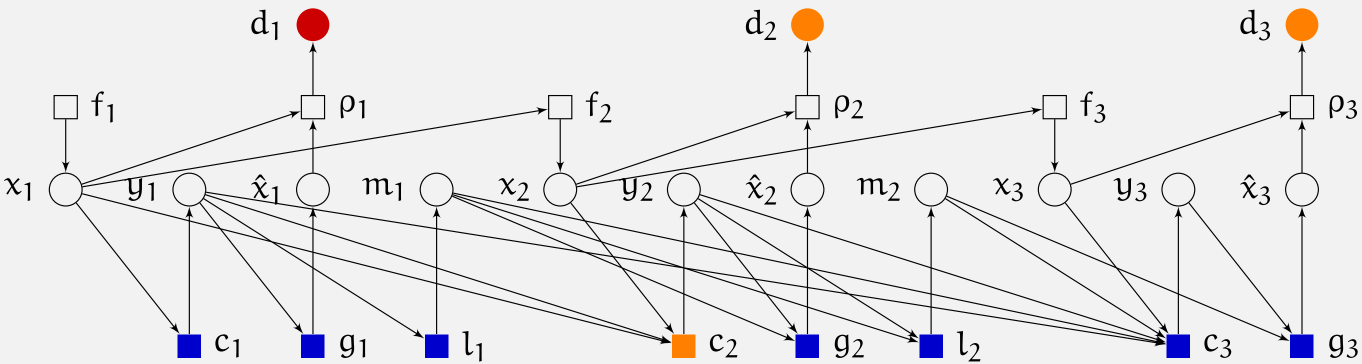
Structural Results for Dec MDP – Step 3



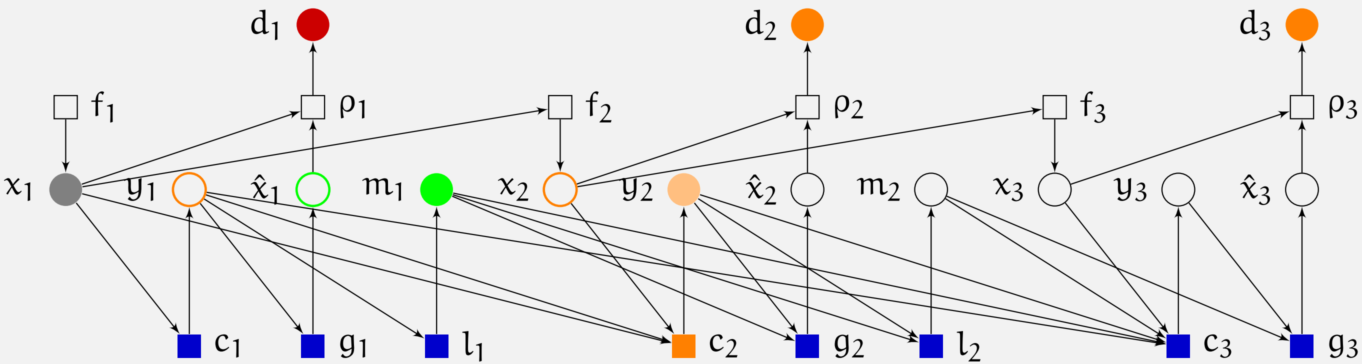
Structural Results for Dec MDP – Step 4



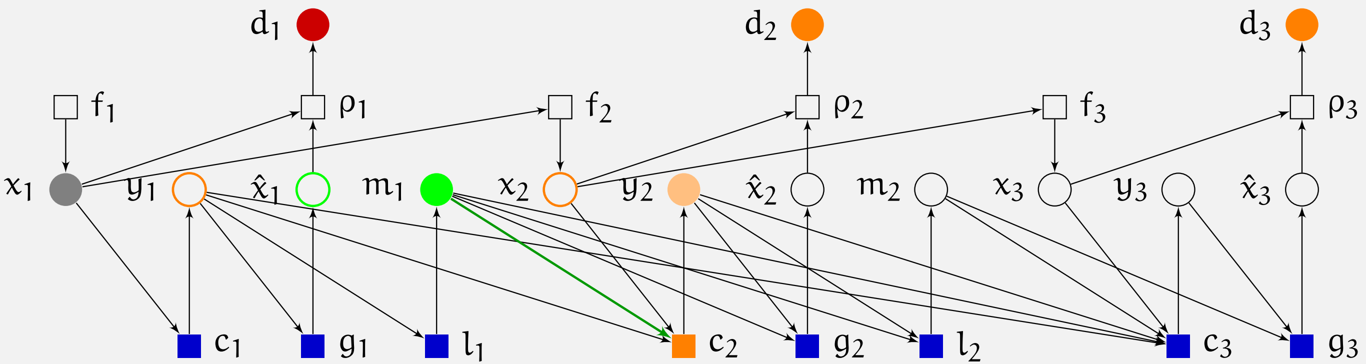
Structural Results for Dec MDP – Step 5



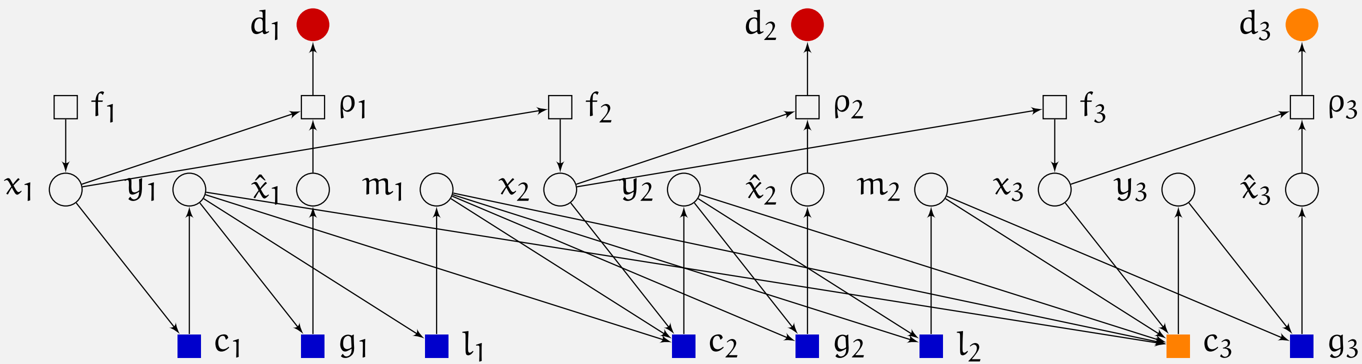
Structural Results for Dec MDP – Step 6



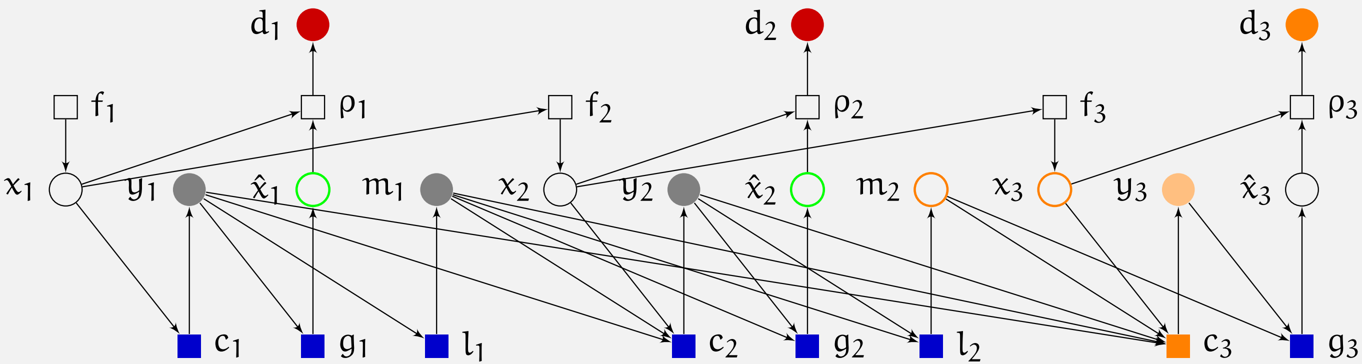
Structural Results for Dec MDP – Step 7



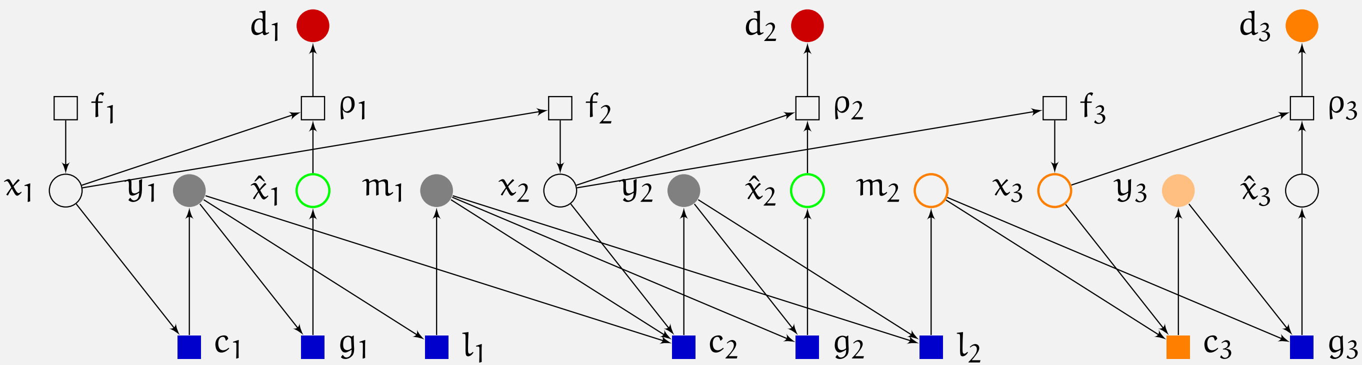
Structural Results for Dec MDP – Step 8



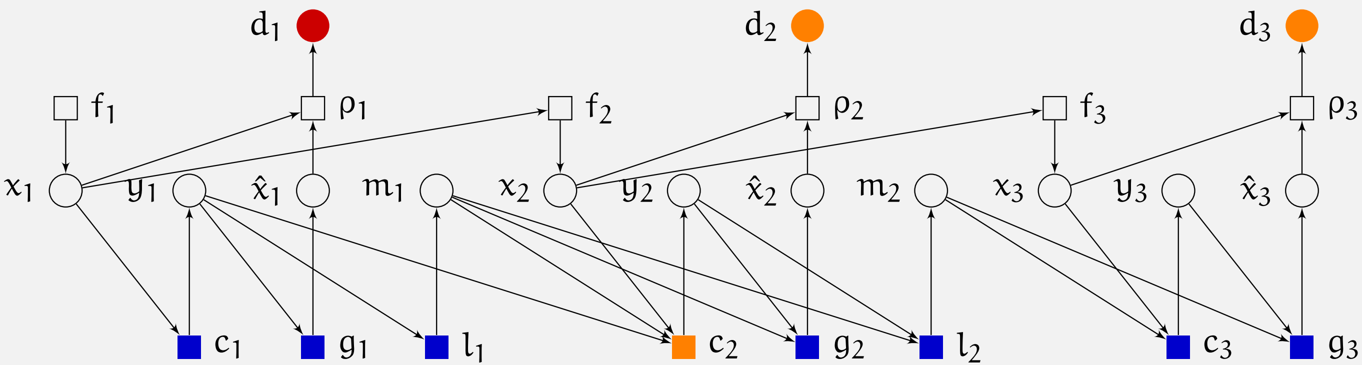
Structural Results for Dec MDP – Step 9



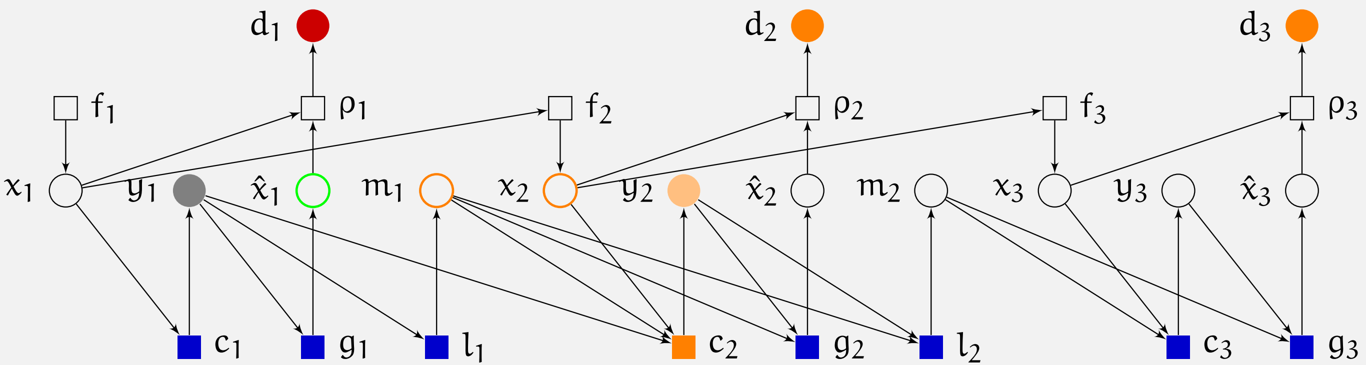
Structural Results for Dec MDP – Step 10



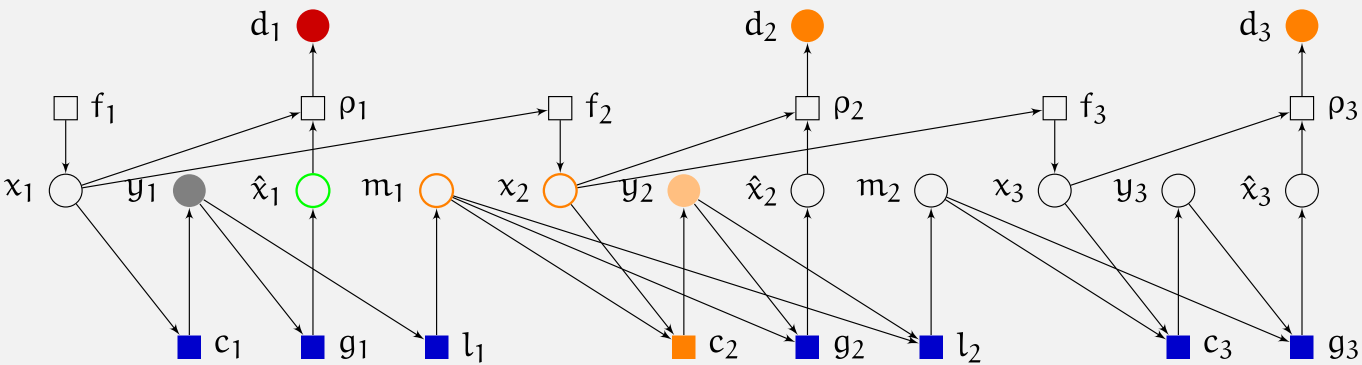
Structural Results for Dec MDP – Step 11



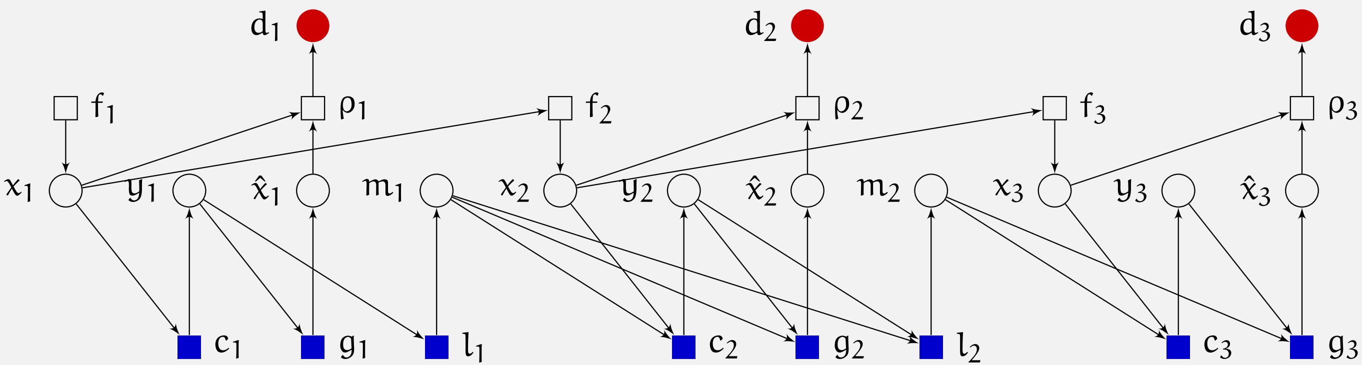
Structural Results for Dec MDP – Step 12



Structural Results for Dec MDP – Step 13

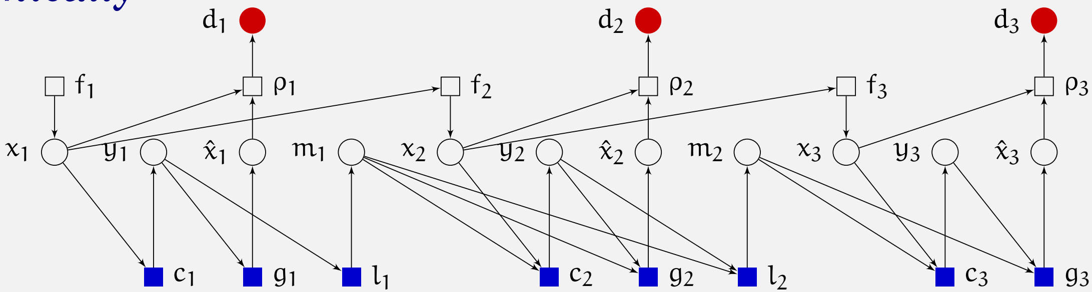


Structural Results for Dec MDP – Step 14



Structural Results for real-time communication

- Graphically



- Mathematically
  - ▷ Original Encoder: $y_t = c_t(x_1, \ldots, x_t, y_1, \ldots, y_{t-1})$
  - ▷ New encoder: $y_t = c_t(x_t, m_{t-1})$

Automated Structural results

- Simplify Once
  - For each control node
    - Find irrelevant nodes and functionally determined nodes.
    - Remove edges from irrelevant nodes, add edges from functionally determined nodes.
- Find fixed point
  - Keep on simplifying until the graph does not change
- Software Implementation
  - A EDSL to find structural results (is there times for a demo??)

Conclusion

Conclusion

- First Model
  - Private and common observations
  - The notion of a coordinator
  - Sequential decomposition for finite and infinite horizon
- Second Model
  - Graphical model
  - Automated structural results

Future directions

- Computational algorithms

  The theory will not be practical until efficient algorithms to solve nested optimality equations are identified

- Graphical Model
  - ▷ Automated method to discover belief states
  - ▷ Automated method for sequential decomposition

- Extensions to non-stochastic systems
  - ▷ Appear feasible
  - ▷ Use σ-algebra rather than the measure induced by the σ-algebra

Thank you