Non-classical information structures

Examples of sequential decomposition

Aditya Mahajan

Yale University

Thanks to

Ashutosh Nayyar and Demosthenis Teneketzis,
University of Michigan

Information Structures

- Information structures :

Collection of data available to each agent to make a decision

- Classical information structure (Centralized Sysmtes)

An agent knows the data available
to all agents that acted earlier

- Non-classical information structure (Decentralized Systems)

Centralized stochastic control – The two models

- Models
  - ▷ LQG ▷ Control of Markov Chains
- Structural results
  - ▷ (LQG) Affine control laws are optimal
  - ▷ (MC) Controls based on belief state are optimal
- Sequential decomposition

  Break a one-shot optimization into a sequence of nested optimizations

  - ▷ Dynamic programming

Outline

- Sequential decomposition in centralized stochastic control
  - MDP (Markov decision processes)
  - POMDP (Partially observable Markov decision processes)
- Sequential decomposition in decentralized stochastic control
  - Common observations
  - No common observation
- Examples
  - Decentralized detection
  - Queueing theory
  - Robotics

Centralized stochastic control – MDP

- Model

$$\begin{aligned}
\text{State Update} &: X(t+1) = f_t(X(t), U(t), W(t)) \\
\text{Observation} &: Y(t) = X(t) \\
\text{Control} &: U(t) = g_t(X(1{:}t), U(1{:}t-1)) \\
\text{Objective} &: E^G\left\{ \sum_{t=1}^{T} c_t(X(t), U(t)) \right\}
\end{aligned}$$

- Structural Result

$$\text{Control} : U(t) = g_t(X(t))$$

- Sequential Decomposition

$$V_t(x(t)) = \min_{u(t)} \left[c_t(x(t), u(t)) + E\left\{ V_{t+1}(X(t+1)) \mid x(t), u(t) \right\} \right]$$

Centralized stochastic control – POMDP

- Model

$$
\begin{aligned}
\text{State Update} &: X(t+1) = f_t(X(t), U(t), W(t)) \\
\text{Observation} &: Y(t) = h_t(X(t), N(t)) \\
\text{Control} &: U(t) = g_t(Y(1{:}t), U(1{:}t-1)) \\
\text{Objective} &: E^G\left\{ \sum_{t=1}^{T} c_t(X(t), U(t)) \right\}
\end{aligned}
$$

- Structural Result

$$
\begin{aligned}
\text{Control} &: U(t) = g_t(B(t)) \\
\text{where} &\quad B(t)(x) = \Pr\left(X(t) = x \,|\, Y(1{:}t), U(1{:}t-1)\right)
\end{aligned}
$$

- Sequential Decomposition

$$
V_t(b(t)) = \min_{u(t)} \left[E\left\{ c_t(x(t), u(t)) + V_{t+1}(B(t+1) \mid b(t), u(t) \right\} \right]
$$

Decentralized Systems

Decentralized stochastic control

- Cost of decentralization
  - Triple objective of control: control, estimation, and communication
- No easy way to find structural results
  - Can consider the problem from the p.o.v. of one agent and fix the strategies of all other agents
  - Works for two agent problems
  - May not work in general and may not give compact structural results
- No general way to obtain sequential decomposition (until now)
  - Coupled dynamic programs can give locally optimal strategies

Decentralized stochastic control

- Our approach (Mahajan, Nayyar, Teneketzis 2008)

  Find sequential decomposition of specific non-trivial instances.

- Two models

  - Model A — Common observations
  - Model B — No common observations

- Cost of decentralization

  - Triple objective of control: control, estimation, and communication
  - Each step of sequential decomposition is a functional optimization problem

Model A – Common observations

- State Update:

$$X(t+1) = f_t(X(t), U(t), W(t)) \quad \text{where } U(t) = [U_1(t), \ldots, U_n(t)]$$

- Common Observations: $Z(t) = c_t(X(t), Q(t))$

- Private Observations: $Y_i(t) = h_{i,t}(X(t), N(t))$

- Control: $U_i(t) = g_{i,t}(Z(1{:}t-1), Y_i(t), M_i(t-1))$

- Memory: $M_i(t) = l_{i,t}(Z(1{:}t-1), Y_i(t), M_i(t-1))$

- Objective: $E^{G,L}\left\{ \sum_{t=1}^{T} c_t(X(t), U(t)) \right\}$

Main Idea:

Think in term of

common agent

Common agent

- Partial functions

$$g_{i,t} : \mathcal{Z}^{t-1} \times \mathcal{Y}_i \times \mathcal{M}_i \to \mathcal{U}_i$$

$$g_{i,t} : \mathcal{Z}^{t-1} \to \big(\mathcal{Y}_i \times \mathcal{M}_i \to \mathcal{U}_i\big)$$

$$l_{i,t} : \mathcal{Z}^{t-1} \times \mathcal{Y}_i \times \mathcal{M}_i \to \mathcal{M}_i$$

$$l_{i,t} : \mathcal{Z}^{t-1} \to \big(\mathcal{Y}_i \times \mathcal{M}_i \to \mathcal{M}_i\big)$$

- Common agent decides partial functions

$$U_i(t) = \hat{g}_{i,t}(Y_i(t), M_i(t-1)) \qquad \text{where} \qquad \hat{g}_{i,t} = \gamma_{i,t}(z(1{:}t))$$

$$M_i(t) = \hat{l}_{i,t}(Y_i(t), M_i(t-1)) \qquad \text{where} \qquad \hat{l}_{i,t} = \lambda_{i,t}(z(1{:}t))$$

Equivalent model

- Augmented State : $\big(X(t), Y(t), M(t-1)\big)$

  where $Y(t) = [Y_1(t), \ldots, Y_n(t)$ and same for $M(t)$.

- State Update

$$X(t+1) = f_t(X(t), U(t), W(t)) \qquad Y_i(t) = h_{i,t}(X(t), N(t))$$

$$U_i(t) = \hat{g}_{i.t}(Y_i(t), M_i(t-1)), \qquad M_i(t) = \hat{l}_{i.t}(Y_i(t), M_i(t-1))$$

- Observations: $\quad Z(t) = c_t(X(t), U(t-1), Q(t))$

- Control

$$\hat{g}_{i,t} = \gamma_{i,t}(Z(1{:}t-1))$$

$$\hat{l}_{i,t} = \lambda_{i,t}(Z(1{:}t-1))$$

- Objective

$$E^{\Gamma,\Lambda}\left\{ \sum_{t=1}^{T} c_t(X(t), Y(t), M(t-1), \hat{g}_{i,t}, \hat{l}_{i,t}) \right\}$$

Equivalent model

Equivalent to a centralized problem!

- Information state

$$\pi(t)(z(1{:}t)) = \Big[\big[[\pi(t)(x, y, m)]\big]\Big]$$

where

$$\pi(t)(x, y, m) = \Pr\left(X(t) = x, Y(t) = y, M(t-1) = m \,|\, Z(1{:}t) = z(1{:}t)\right)$$

- Sequential decomposition

$$V_t(\pi(t)) = \min_{\hat{g}(t), \hat{l}(t)} \left[E\left\{c_t(x(t), u(t)) + V_{t+1}(\Pi(t+1) \mid \pi(t), \hat{g}(t), \hat{l}(t)\right\}\right]$$

Model B – No common observations

- State Update:

$$X(t+1) = f_t(X(t), U(t), W(t)) \quad \text{where } U(t) = [U_1(t), \dots, U_n(t)]$$

- Observations: $\quad Y_i(t) = h_{i,t}(X(t), N(t))$

- Control: $\quad U_i(t) = g_{i,t}(Y_i(t), M_i(t-1))$

- Memory: $\quad M_i(t) = g_{i,t}(Y_i(t), M_i(t-1))$

- Objective: $\quad E^{G,L}\left\{ \sum_{t=1}^{T} c_t(X(t), U(t)) \right\}$

Main Idea:

Think in term of

system designer

Equivalent model

- Augmented State : $\big(X(t), Y(t), M(t-1)\big)$

  where $Y(t) = [Y_1(t), \ldots, Y_n(t)$ and same for $M(t)$.

- State Update

$$X(t+1) = f_t(X(t), U(t), W(t)) \qquad Y_i(t) = h_{i,t}(X(t), N(t))$$

$$U_i(t) = g_{i.t}(Y_i(t), M_i(t-1)), \qquad M_i(t) = l_{i.t}(Y_i(t), M_i(t-1))$$

- Observations: Nothing

- Control

$$g_{i,t} = \gamma_{i,t}(g(t-1), l(t-1))$$

$$l_{i,t} = \lambda_{i,t}(g(t-1), l(t-1))$$

- Objective

$$E^{\Gamma,\Lambda}\left\{\sum_{t=1}^{T} c_t(X(t), Y(t), M(t-1), g_{i,t}, l_{i,t})\right\}$$

Equivalent model

Equivalent to a centralized problem!

- Information state

$$\pi(t)(z(1{:}t)) = \left[\left[[\pi(t)(z(1{:}t))(x, y, m)]\right]\right]$$

where

$$\pi(t)(z(1{:}t))(x, y, m) = \Pr\left(X(t) = x, Y(t) = y, M(t-1) = m\right)$$

- Sequential decomposition

$$V_t(\pi(t)) = \min_{g(t), l(t)} \left[E\left\{c_t(x(t), u(t)) + V_{t+1}(\Pi(t+1) \mid \pi(t), g(t), l(t)\right\}\right]$$

Non-classical information structures:

Sequential decomposition is possible

Are these models useful – Special Cases

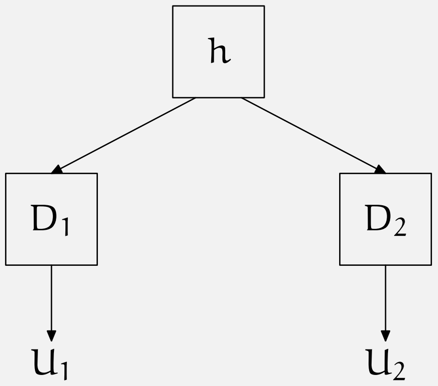
- k-step state sharing information structure
- k-step control sharing information structure
- k-step observation sharing information structure
- Witsenhausen's general sequential team model
- Witsenhausen's standard form

Are these models useful – Examples

- Decentralized Detection
  - Decentralized Wald Problem
  - Wald problem with 1-bit memory
- Control of Queues
  - Multi-access broadcast
  - Load balancing in queues
- Robotics
  - Robot rendezvous with communications
  - Decentralized task scheduling
- Toy problem
  - Decentralized tiger problem

Decentralized Wald problem

- Decisions = { More meas., decide H_0, decide H_1 }
- action = fn(all past observations)
- Each measurement by each sensor costs c
- Final cost $J(u_1, u_2, h) = \begin{cases} 0 & \text{if } u_1 = u_2 = h, \\ 1 & \text{if } u_1 \neq u_2 \\ k & \text{if } u_1 = u_2 \neq h \end{cases}$
- Minimize $E\left\{c\tau_1 + c\tau_2 + J(u_1, u_2, h)\right\}$



Decentralized Wald problem

- Studied by Teneketzis and Ho, 1985
- Obtained structural results:

> Optimal to take action based on
> likelihood ratio test (threshold rule)
>
> $$\text{action} = \text{fn}(\text{posterier of } \mathrm{H}_0)$$

- Obtained coupled dynamic programs to determine locally optimally thresholds.

Globally optimal thresholds – Use Model B

Structural results make the model equivalent to Model B

- Define: $\pi(t) = \Pr\left(H, U_1(t), U_2(t), \lambda_1(t), \lambda_2(t)\right)$ or equivalently
  $\pi(t) = \Big[\Pr\left(U_1(t), U_2(t), \lambda_1(t), \lambda_2(t) \mid H_0\right), \quad \Pr\left(U_1(t), U_2(t), \lambda_1(t), \lambda_2(t) \mid H_1\right)\Big]$

- Sequential Decomposition

$$V_t(\pi(t)) = \min_{\alpha(t),\beta(t)} \Big[E\Big\{c_t(h, u_1(t), u_2(t)) + V_{t+1}(\Pi(t+1) \mid \pi(t), \alpha(t), \beta(t)\Big\}\Big]$$

Wald problem with 1 bit of memory

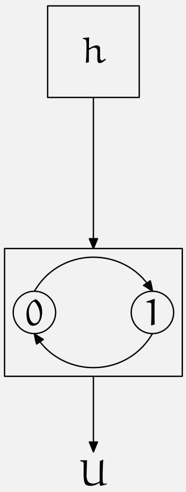
- Decisions = { More meas. and update memory, decide H_0, decide H_1 }

- Each measurement by each sensor costs c

- (action, next memory) = fn(current observation, current memory)

- Final cost $J(u,h) = \begin{cases} 0 & \text{if } u = h, \\ L_0 & \text{if } u \neq h = H_0 \\ L_1 & \text{if } u \neq h = H_1 \end{cases}$

- Minimize $E\{c\tau + J(u,h)\}$



Wald problem with 1 bit of memory

- Studied by Sandell, 1974
- Presents a sequential decomposition attributed to Witsenhausen

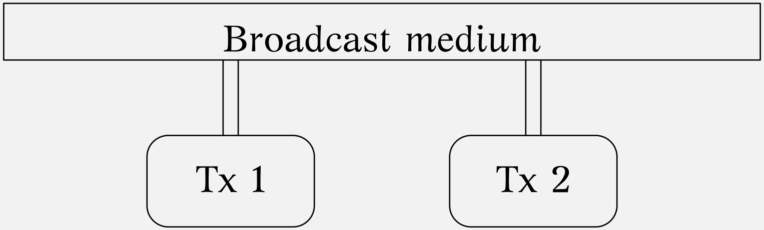
Globally optimal design – Use Model B

- Define: $\pi(t) = \Big[\Pr\left(M(t)\,|\,H_0\right), \quad \Pr\left(M(t)\,|\,H_1\right)\Big]$

- Sequential Decomposition

$$V_t(\pi(t)) = \min \begin{Bmatrix} E\{\text{cost of stopping} \mid \pi(t), u = H_0\}, \\ E\{\text{cost of stopping} \mid \pi(t), u = H_1\}, \\ c + E\{V_{t+1}(\Pi(t+1)) \mid \pi(t), g(t)\} \end{Bmatrix}$$

Multi-access broadcast



- Independent Bernoulli arrivals
- One user transmits $\Longrightarrow$ successful transmission
- Both users transmit $\Longrightarrow$ packet collision
- Channel feedback available to both users
- action = fn(all past queue sizes, all past feedback)
- Objective: Maximize throughput or minimize delay

Multi-access broadcast

- Studied by Hluchyj and Gallager, 1981
- Restricted attention to “window protocols” and symmetric arrival rates

Determining optimal design – Use Model A

- Structural results

action = fn(current queue size, all feedback)

- Define : $\pi(t) = \Pr\left(\text{queue size of user } 1, \text{queue size of user } 2 \,|\, \text{all past feedback}\right)$

- Sequential decomposition

$$V_t(\pi(t)) = \min_{g_1(t), g_2(t)} \Big[\mathbb{E}\big\{ c(x(t), u(t)) + V_{t+1}(\Pi(t+1)) \mid \pi(t), g_1(t), g_2(t) \big\} \Big]$$

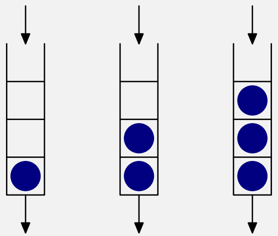
Multi-access broadcast

- For symmetric arrival rates and buffer size 1, the sequential decomposition can be used to prove that Hluchyj-Gallager strategy is optimal among all strategies.

- Structural results can be further simplified (Anastasopoulos, preprint)

$$\pi(t) = \big(\,\Pr\left(\text{queue size of user } 1 \,|\, \text{all past feedback}\right),\\ \Pr\left(\text{queue size of user } 2 \,|\, \text{all past feedback}\right)\big)$$

Load balancing in Queues



- Independent Bernoulli arrivals at rate λ_i
- Geometric service times μ_i
- Each queue can look at the queue sizes of its neighbors and transfer packets to neighbors.
- Finite buffer.
- Waiting cost w, dropping cost d, switching cost c
- action = fn(all past queue lengths of self and neighbors)
- Objective: Minimize expected waiting plus switching cost

Load balancing in Queues

- Considered by Cogill, Rotkowitz, Roy, and Lall, 2004
- Assumed each queue uses only the current queue lengths to determine its action.

  action = fn(current queue lengths of self and neighbors)

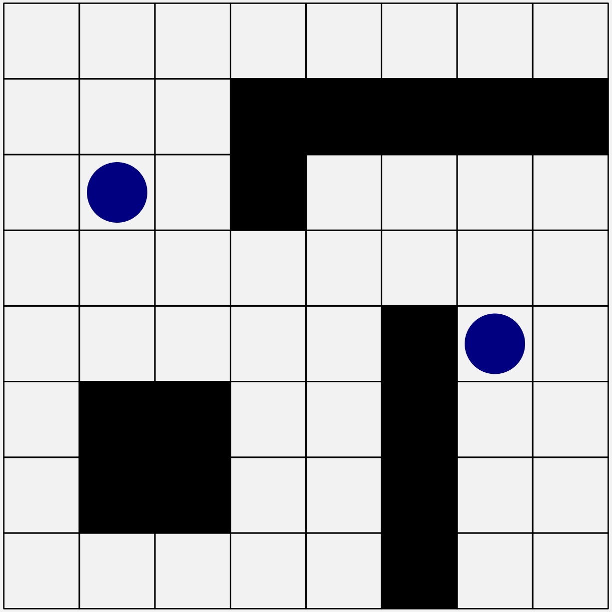
- Provided an approximate dynamic programming approach

Determining optimal design – Use model B

- Define: $\pi(t) = \Pr\left(\text{queue length of all users}\right)$
- Sequential Decomposition

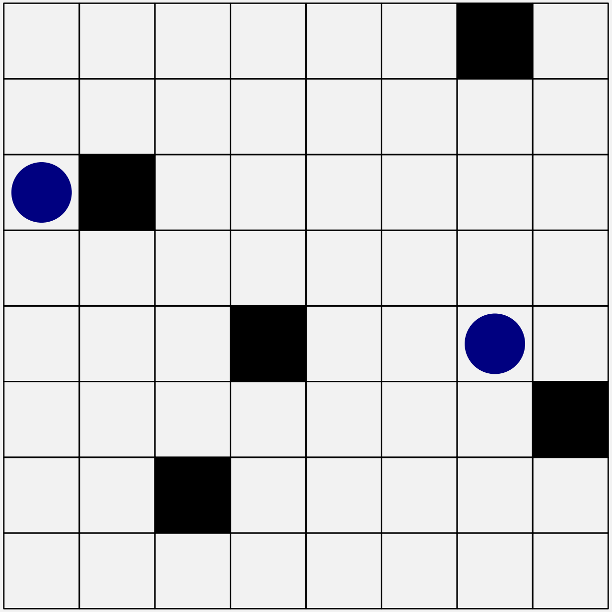
$$V_t(\pi(t)) = \min_{g(t)} \left[E\left\{ c(x(t), u(t)) + V_{t+1}(\Pi(t+1)) \mid \pi(t), g(t) \right] \right.$$

Robot rondevouz



Bernsein, Zibersein, Immerman, 2000

Robot task sharing



Conclusion

Finding structural results is an art

Obtaining a sequential decomposition

can be turned into a science