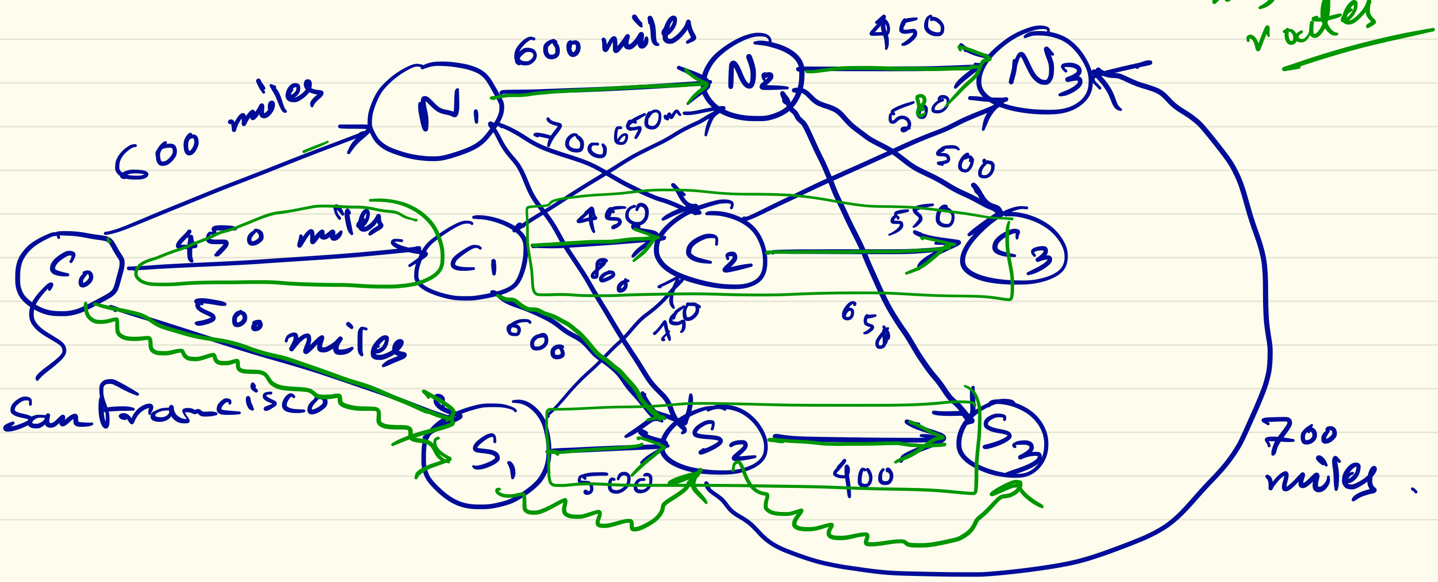
Lecture #15

Dynamic Programming (we will start with discrete time)

Network of roads in US

Want to find the shortest drive from SF to Chicago.

all 1 way routes



- From N_2, shortest path to East Cost is 450 miles.

450 N_2

0 N_3

0 C_3

0 S_3

-

	1050 N_1	450 N_2	0 N_3
1400 C_0	1000 C_1	550 C_2	0 C_3
$(C_0 \to S_1)$	900 S_1	400 S_2	0 S_3
$(S_1 \to S_2)$	$(N_1 \to N_2)$ $(C_1 \to C_2)$	$(N_2 \to N_3,\ C_2 \to N_3,\ S_2 \to S_3)$	

Observations:

① In order to solve the problem for one initial condition, we had to solve for all starting states/vertices/cities.

↓

This procedure is called Dynamic Programming (DP)

↓

Hence computationally (exponential complexity) difficult.

② DP gives you a "closed-loop-policy"/ "feedback policy"

$\Leftrightarrow$ actions (controls) as $f^{\underline{n}}$ of state you're in now

[If ever I find myself in Denver then go south].

③ Different from Open-loop policy (time-table)
(close your eyes, drive 3 hours east, then take left turn)

④ DP gives closed-loop policy.

⑤ DP proceeds Backward in time
($\Leftarrow$ "Backward Recursion")

Solve for t days remaining

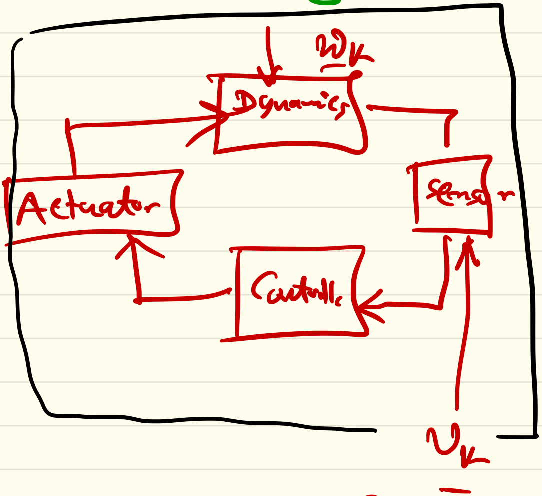
$\downarrow$

Then (t+1) days remaining etc.

⑥ Segments / subarcs of optimal path are themselves optimal
($\Leftrightarrow$ Bellman's Principle of Optimality)

7 Recursion :

Optimal cost with $(t+1)$ days remaining $=$ $\min\limits_{\{\text{set of actions}\}}$ $\left[\text{Immediate cost of my action} + \text{Optimal cost from where that cost takes you} \right]$



To fix ideas, consider discrete time.

$$\underline{x}_{k+1} = \underline{f}_k(\underline{x}_k, \underline{u}_k, \underline{w}_k)$$

$$\underline{y}_k = \underline{h}_k(\underline{x}_k, \underline{u}_k, \underline{v}_k)$$

- $\underline{x}_k \in \mathcal{X} \subset \mathbb{R}^n$ (state space), $\underline{u}_k \in \mathcal{U}$ (control space)
- For each $k = 0, 1, 2, \ldots,$ the control values $u_k \in \mathcal{U} \subset \mathbb{R}^m$.

A feasible control law is a sequence of policies:

Remember: $\boxed{\text{Policy/law/ feedback} \neq \text{Action/ Control}}$

Policy/ law/feedback :

$\underline{\gamma} = \{ \underline{\gamma}_0, \underline{\gamma}_1, \underline{\gamma}_2, \ldots \}$ s.t. $\underline{u}_k = \underline{\gamma}_k (\underline{y}_k)$

$\in \mathcal{U}$

Feedback @ time 0

Feedback @ time 1

$\gamma(k, y_k)$

- Let Γ be the set of all possible policies :

$$\boxed{u(t) = \underline{\gamma}(\underline{x}(t), t)}$$

- We wish to find the best $\underline{\gamma}$ in Γ.

$\therefore$ We need criterion to compare different policies

we associate cost for each policy, and declare the best one is the one that minimizes cost.

- Our cost function:

$$J(\underline{\gamma}) := \underbrace{c_T\left(\underline{x}(T)\right)}_{\substack{\| \\ \phi(x(T),\,T) \\ \text{(terminal cost)}}} + \sum_{k=0}^{T-1} \underbrace{c_k(\underline{x}_k, \underline{u}_k)}_{\substack{\| \\ L(k,\,\underline{x}_k,\,\underline{u}_k) \\ \text{(Lagrangian)}}}$$

- Finite horizon: $T < \infty$

$\Updownarrow$

control law is a finite policy sequence:

$$\underline{\gamma} = \{\underline{\gamma}_0, \underline{\gamma}_1, \ldots, \underline{\gamma}_{T-1}\}$$

- The term $c_k(\underline{x}_k, \underline{u}_k)$ is called immediate/ one period cost.

Our plan :

Stochastic Dynamic Programming

MDP (Markov Decision Processes)

(& its generalization POMDP (Partially observed Markov Decision Processes))

Deterministic DP

Discrete time

Generalize this story in continuous time.

Stochastic DP

- Deterministic DP is special case: $\underbrace{\underline{w}_k \equiv \underline{0}}_{\text{process noise}}$, $\underline{v}_k \equiv \underline{0}$

measurement noise

- Stochastic DP:

$\underline{w}_k \in \mathcal{W}$, $\underline{v}_k \in \mathcal{V}$;

random vectors realized from (Discrete time stochastic process)

$$\mathcal{W} := (\mathbb{P}_w, \Omega_w, F_w)$$

$$\mathcal{V} := (\mathbb{P}_v, \Omega_v, F_v)$$

Then $\underline{x}_k, \underline{u}_k$ are random vectors, and hence $J(\underline{\gamma})$ is a random variable

(i.e.) $J(\underline{\gamma}) \equiv J(\underline{\omega}, \underline{\gamma})$, $\underline{\omega} \in \Omega_w \times \Omega_v$

sample path index

To resolve sample path dependency, we take

$$J(\gamma) = \mathbb{E}[J(\underline{\gamma})]$$

Let $\underline{\gamma}^* = \underset{\underline{\gamma} \in \Gamma}{\operatorname{argmin}} \ \mathbb{E}[J(\underline{\gamma})]$

and $\underbrace{J^*}_{\substack{\text{Deterministic} \\ \text{scalar} \geq 0}} = \min_{\underline{\gamma} \in \Gamma} \mathbb{E}[J(\underline{\gamma})]$.

We say $\underline{\gamma}^*$ is optimal policy, J^* is optimal cost.

We will now focus on : MDP
(Markov Decision Process)

Complete information/Fully Observed case :

$$\underline{y}_k \equiv \underline{x}_k ,$$

$$\underline{x}_{k+1} = f_k(\underline{x}_k, \underline{u}_k, \underline{w}_k) ,$$

$\underline{x}_k \in \mathcal{X} \subset \mathbb{R}^n$
$\underline{u}_k \in \mathcal{U} \subset \mathbb{R}^m$
$\underline{w}_k \in \mathcal{W} \subset \mathbb{R}^p$

Let $\underline{u}_k = \underline{\gamma}_k(\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_k)$
is allowed to depend on
previous states.
(i.e) $\underline{\gamma}_k$ is history-dependent
policy.

More generally, History upto time $t =: H_t$
$:= \{\underline{x}_0, \underline{u}_0, \underline{x}_1, \underline{u}_1, \ldots, \underline{x}_{k-1}, \underline{u}_{k-1}, \underline{x}_k\}$

At each ~~k~~, $\gamma_k(H_k) = \underline{u}_k$

$\gamma_k : H_k \mapsto \mathcal{U}$

$\underline{\gamma} = (\underline{\gamma}_0, \underline{\gamma}_1, \ldots, \gamma_{T-1})$ is called history-dependent policy

H_k is history up until time k.

History Dependent Policies

Randomized

$\gamma_k : H_k \mapsto \mathbb{P}$rob. over $\mathcal{U}$

(Choose u_k as a sample from that Probability)

Non-randomized

$\gamma_k : H_k \mapsto \underline{\mathcal{U}}$

returns u_k (particular action)

<u>Detour</u>: Markov process:

$$\mathbb{P}(\text{Future} \mid \text{Past \& Present})$$
$$= \mathbb{P}(\text{Future} \mid \text{Present})$$

Another way to write:

$$\mathbb{P}(\text{Past \& Future} \mid \text{Present})$$
$$= \mathbb{P}(\text{Past} \mid \text{Present})\, \mathbb{P}(\text{Future} \mid \text{Past \& Present})$$
$$= \mathbb{P}(\text{Past} \mid \text{Present})\, \mathbb{P}(\text{Future} \mid \text{Present}) \quad \text{-----} (A)$$

($\because$ Markov)

($\because \mathbb{P}(A, B) = \mathbb{P}(A)\,\mathbb{P}(B \mid A)$)

One way to think this is to recall:

if $\mathbb{P}(A \& B) = \mathbb{P}(A)\,\mathbb{P}(B)$

then A & B are independent.

So (*) means:
" Past & future are conditionally independent, given the present".

can be taken as alternative defn of Markov process

Discrete Time: Markov Chain (have states $\{s_1, \dots, s_m\}$)

$$\mathbb{P}\left(\underline{x}(t+1) = s_j \,\middle|\, \underline{x}(t) = s_i\right)$$

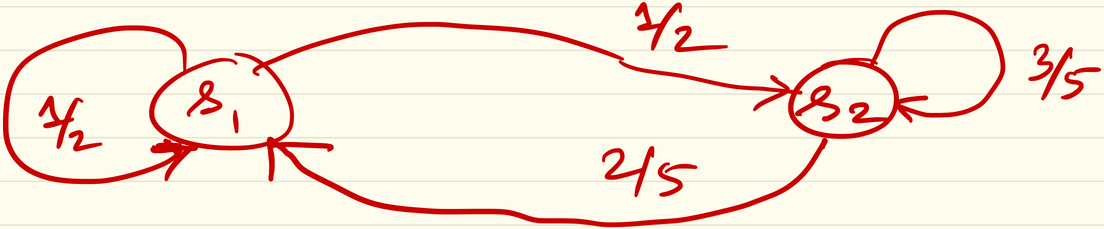
$$= p_{ij} \in [0,1]$$

This defines an $m \times m$ matrix

$\underset{\sim}{P} = [p_{ij}]$ where $0 \leq p_{ij} \leq 1$, & $\sum_{j=1}^{m} p_{ij} = 1$

called (row) stochastic matrix

Example: 2 state Markov Chain:



$$\therefore \quad P = \begin{bmatrix} 1/2 & 1/2 \\ 2/5 & 3/5 \end{bmatrix}$$

Example: 3 state Markov Chain

$\{S, C, R\}$

$$P = \begin{array}{c|ccc} & S & C & R \\ \hline S & 1/2 & 1/2 & 0 \\ C & 1/2 & 1/4 & 1/4 \\ R & 0 & 1/2 & 1/2 \end{array}$$

$\rightarrow$ sums to 1.

etc.

Coming back to feedback policy:

Def$^{\underline{n}}$: A feasible policy γ_k is called "Markovian" or "Markov Policy" if $\underline{\gamma_k}$ only depends on $\underline{y_k} \equiv \underline{x_k}$ (MDP)

("action now" depends on "state now")

set of Markov policies: $\Gamma_M \subset \underline{\Gamma}$

all history dependent randomized policies

$$\boxed{\text{Intuition suggests: } \quad \underline{\gamma}^* \in \Gamma_M .}$$

Dynamic Programming Sol$^{\underline{n}}$:

Let $V_k(\underline{x})$ = Optimal remaining expected cost from state $\underline{x}$ at time k.
(generic)

$$= \inf_{(\underline{\gamma}_k, \underline{\gamma}_{k+1}, \ldots, \underline{\gamma}_{T-1})} \mathbb{E}\left[\left\{c_T(\underline{x}(T)) + \sum_{s=k}^{T-1} c_s(\underline{x}_s, \underline{u}_s)\right\} \middle| \underline{x}_k = \underline{x}\right]$$

Under a Markovian policy, can show that:

$$V_k^{\underline{\gamma}}(\underline{x}_k^{\underline{\gamma}}) = \mathbb{E}\left[\text{copy} \,\middle|\, \underline{x}_k^{\underline{\gamma}}\right]$$

(we're using: If $\underline{\gamma} \in \Gamma_M$, then $\{\underline{x}_k^{\underline{\gamma}}\}$ is a Markov process)

Define:

$$V_T(\underline{x}) := c_T(\underline{x})$$

Nothing random here

and $V_k(\underline{x}) :=$

$$\inf_{u(\cdot) \in \mathcal{U}} \left\{ c_k(\underline{x}, \underline{u}) + \mathbb{E}_{\underline{w}_k}\left[V_{k+1}(f_k(\underline{x}, \underline{u}, \underline{w}_k)) \right] \right\},$$

where $k = T-1, T-2, \ldots, 0$

This minimization is over actions (NOT over policies)

(Even if policies are randomized, there is nothing random about this minimization)

$$\underset{\gamma(\cdot)}{\text{minimize}} \; \mathbb{E}\left[C_T(\underline{x}(T)) + \sum_{k=0}^{T-1} c_k(\underline{x}_k, u_k) \right]$$