

6.231 DYNAMIC PROGRAMMING

LECTURE 12

LECTURE OUTLINE

- Average cost per stage problems
- Connection with stochastic shortest path problems
- Bellman's equation
- Value iteration
- Policy iteration

AVERAGE COST PER STAGE PROBLEM

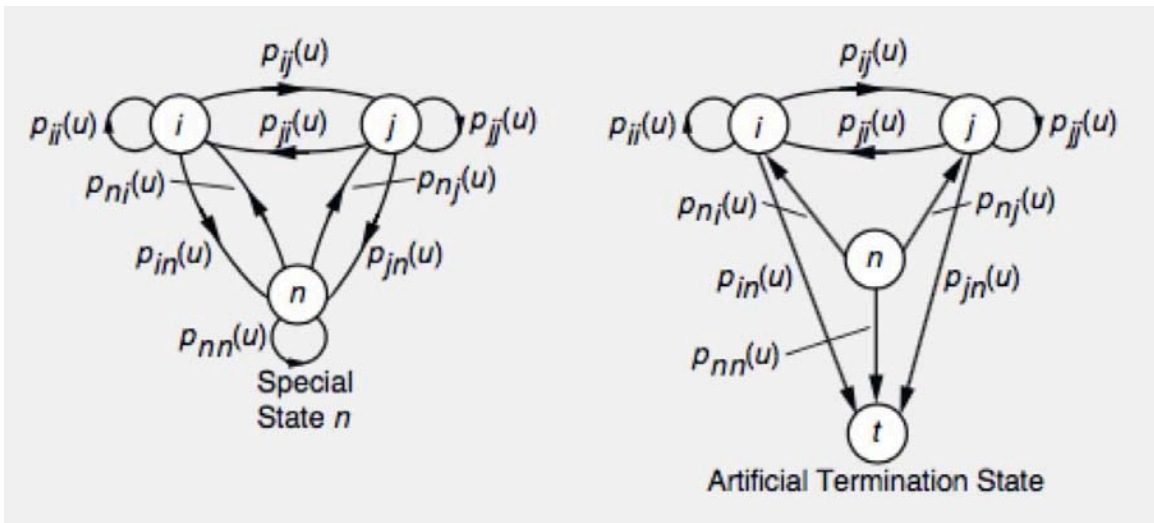
- Assume a stationary system with finite number of states and controls.
- Minimize over policies $\pi = \{\mu_0, \mu_1, \dots\}$

$$J_\pi(x_0) = \lim_{N \rightarrow \infty} \frac{1}{N} \underset{w_k}{E}_{k=0,1,\dots} \left\{ \sum_{k=0}^{N-1} g(x_k, \mu_k(x_k), w_k) \right\}$$

- Important characteristics (not shared by other types of infinite horizon problems).
 - For any fixed T , **the cost incurred up to time T does not matter** (only the state that we are at time T matters)
 - **If all states “communicate” the optimal cost is independent of initial state** [if we can go from i to j in finite expected time, we must have $J^*(i) \leq J^*(j)$]. So $J^*(i) \equiv \lambda^*$ for all i .
 - Because “communication” issues are so important, the methodology relies heavily on Markov chain theory.
 - **The theory depends a lot on whether the chains corresponding to policies have a single or multiple recurrent classes.** We will focus on the simplest version, using SSP theory.

CONNECTION WITH SSP

- **Assumption:** State n is special, in that for all initial states and all policies, n will be visited infinitely often (with probability 1).
- Then we expect that $J^*(i) \equiv \text{some } \lambda^*$
- Divide the sequence of generated states into cycles marked by successive visits to n .
- **Let's focus on a single cycle:** It can be viewed as a state trajectory of an SSP problem with n as the termination state.



- Let the cost at i of the SSP be $g(i, u) - \lambda^*$
- We will argue (informally) that

Av. Cost Probl. \equiv A Min Cost Cycle Probl. \equiv SSP Probl.

CONNECTION WITH SSP (CONTINUED)

- Consider a **minimum cycle cost problem**: Find a stationary policy μ that minimizes the **expected cost per transition within a cycle**

$$\frac{C_{nn}(\mu)}{N_{nn}(\mu)},$$

where for a fixed μ ,

$C_{nn}(\mu) : E\{\text{cost from } n \text{ up to the first return to } n\}$

$N_{nn}(\mu) : E\{\text{time from } n \text{ up to the first return to } n\}$

- Intuitively, $C_{nn}(\mu)/N_{nn}(\mu) =$ average cost of μ , and optimal cycle cost $= \lambda^*$, so

$$C_{nn}(\mu) - N_{nn}(\mu)\lambda^* \geq 0,$$

with equality if μ is optimal.

- Consider **SSP with stage costs** $g(i, u) - \lambda^*$. The cost of μ starting from n is $C_{nn}(\mu) - N_{nn}(\mu)\lambda^*$, so **the optimal/min cycle μ is also optimal for the SSP**.
- Also: **Optimal SSP cost starting from $n = 0$** .

BELLMAN'S EQUATION

- Let $h^*(i)$ the optimal cost of this SSP problem when starting at the nontermination states $i = 1, \dots, n$. Then $h^*(1), \dots, h^*(n)$ solve uniquely the corresponding Bellman's equation

$$h^*(i) = \min_{u \in U(i)} \left[g(i, u) - \lambda^* + \sum_{j=1}^{n-1} p_{ij}(u) h^*(j) \right], \forall i$$

- If μ^* is an optimal stationary policy for the SSP problem, we have

$$h^*(n) = C_{nn}(\mu^*) - N_{nn}(\mu^*)\lambda^* = 0$$

- Combining these equations, we have

$$\lambda^* + h^*(i) = \min_{u \in U(i)} \left[g(i, u) + \sum_{j=1}^n p_{ij}(u) h^*(j) \right], \forall i$$

$$h^*(n) = 0$$

- If $\mu^*(i)$ attains the min for each i , μ^* is optimal.
- There is also Bellman Eq. for a single policy μ .

MORE ON THE CONNECTION WITH SSP

- Interpretation of $h^*(i)$ as a **relative or differential cost**: It is the minimum of

$E\{\text{cost to reach } n \text{ from } i \text{ for the first time}\}$

– $E\{\text{cost if the stage cost were } \lambda^* \text{ and not } g(i, u)\}$

- **Algorithms**: We don't know λ^* , so we can't solve the average cost problem as an SSP problem. But similar value and policy iteration algorithms are possible, and will be given shortly.

- **Example**: A manufacturer at each time
 - Receives an order with prob. p and no order with prob. $1 - p$.
 - May process all unfilled orders at cost $K > 0$, or process no order at all. The cost per unfilled order at each time is $c > 0$.
 - Maximum number of orders that can remain unfilled is n .
 - Find a processing policy that minimizes the total expected cost per stage.

EXAMPLE (CONTINUED)

- State = number of unfilled orders. State 0 is the special state for the SSP formulation.

- **Bellman's equation:** For states $i = 0, 1, \dots, n-1$

$$\lambda^* + h^*(i) = \min \left[K + (1 - p)h^*(0) + ph^*(1), \right. \\ \left. ci + (1 - p)h^*(i) + ph^*(i + 1) \right],$$

and for state n

$$\lambda^* + h^*(n) = K + (1 - p)h^*(0) + ph^*(1)$$

Also $h^*(0) = 0$.

- **Optimal policy:** Process i unfilled orders if

$$K + (1 - p)h^*(0) + ph^*(1) \leq ci + (1 - p)h^*(i) + ph^*(i + 1)$$

- Intuitively, $h^*(i)$ is monotonically nondecreasing with i (interpret $h^*(i)$ as optimal costs-to-go for the associate SSP problem). So a **threshold policy is optimal**: process the orders if their number exceeds some threshold integer m^* .

VALUE ITERATION

- **Natural VI method:** Generate optimal k -stage costs by DP algorithm starting with any J_0 :

$$J_{k+1}(i) = \min_{u \in U(i)} \left[g(i, u) + \sum_{j=1}^n p_{ij}(u) J_k(j) \right], \quad \forall i$$

- **Convergence:** $\lim_{k \rightarrow \infty} J_k(i)/k = \lambda^*$ for all i .
- **Proof outline:** Let J_k^* be so generated starting from the opt. differential cost, i.e., the initial condition $J_0^* = h^*$. Then, by induction,

$$J_k^*(i) = k\lambda^* + h^*(i), \quad \forall i, \quad \forall k.$$

On the other hand,

$$|J_k(i) - J_k^*(i)| \leq \max_{j=1, \dots, n} |J_0(j) - h^*(j)|, \quad \forall i$$

since $J_k(i)$ and $J_k^*(i)$ are optimal costs for two k -stage problems that differ only in the terminal cost functions, which are J_0 and h^* .

RELATIVE VALUE ITERATION

- The VI method just described has two drawbacks:
 - Since typically some components of J_k diverge to ∞ or $-\infty$, calculating $\lim_{k \rightarrow \infty} J_k(i)/k$ is numerically cumbersome.
 - The method will not compute a corresponding differential cost vector h^* .
- We can bypass both difficulties by subtracting a constant from all components of the vector J_k , so that the difference, call it h_k , remains bounded.
- **Relative VI algorithm:** Pick any state s , and iterate according to

$$h_{k+1}(i) = \min_{u \in U(i)} \left[g(i, u) + \sum_{j=1}^n p_{ij}(u) h_k(j) \right]$$
$$- \min_{u \in U(s)} \left[g(s, u) + \sum_{j=1}^n p_{sj}(u) h_k(j) \right], \quad \forall i$$

- **Convergence:** We can show $h_k \rightarrow h^*$ (under an extra assumption; see Vol. II).

POLICY ITERATION

- At iteration k , we have a stationary μ^k .
- **Policy evaluation:** Compute λ^k and $h^k(i)$ of μ^k , using the $n + 1$ equations $h^k(n) = 0$ and

$$\lambda^k + h^k(i) = g(i, \mu^k(i)) + \sum_{j=1}^n p_{ij}(\mu^k(i)) h^k(j), \quad \forall i$$

- **Policy improvement:** (For the λ^k -SSP) Find

$$\mu^{k+1}(i) = \arg \min_{u \in U(i)} \left[g(i, u) + \sum_{j=1}^n p_{ij}(u) h^k(j) \right], \quad \forall i$$

- If $\lambda^{k+1} = \lambda^k$ and $h^{k+1}(i) = h^k(i)$ for all i , stop; otherwise, repeat with μ^{k+1} replacing μ^k .
- **Result:** For each k , we either have $\lambda^{k+1} < \lambda^k$ or we have policy improvement for the λ^k -SSP:

$$\lambda^{k+1} = \lambda^k, \quad h^{k+1}(i) \leq h^k(i), \quad i = 1, \dots, n.$$

The algorithm terminates with an optimal policy.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.231 Dynamic Programming and Stochastic Control
Fall 2015

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.