Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

Telecom Paris, 27 November 2020

Outline of the Course: Part I

Michèle Wigger (3C58) and Mustapha Hamad (3C54)

- Markov Chains
- Dynamic Programming for Finite Horizon and Shortest-Paths Problems
- Dynamic Programming for Infinite Horizon Problems with Discounted and Average Cost Functions
- Constrained Markov Decision Processes: Solutions and Suboptimal Policies
- 2 TDs and 1 TP

Outline of the Course: Part II

Mireille Sarkiss, Telecom SudParis, 3C56

- Markov Decision Processes without known transition probabilities
- Reinforcement Learning: exploration/exploitation tradeoff
- Epsilon Greedy, Boltzman Algorithm
- Deep reinforcement learning

Lecture 1 – Finite-State Markov Chains

Definitions and Types of Markov Chains

Definition (First-order Markov Chain)

A stochastic process $\{X_k\}_{k\geq 0} = \{X_0, X_1, X_2, \ldots,\}$ over an alphabet $\mathcal{X}$ is called a (first-order) Markov chain if for all $k = 1, 2, \ldots,$:

$$P_{X_k|X_{k-1},X_{k-2},\ldots,X_0}(a|b,c,\ldots,z) = P_{X_k|X_{k-1}}(a|b), \quad \forall a,b,c,\ldots,z \in \mathcal{X}.$$

- Examples: Random walk, memoryless process, ...
- Statistics of the stochastic process $\{X_k\}_{k\geq 0}$ is determined by P_{X_0} and $\{P_{X_k|X_{k-1}}\}_{k\geq 1}$. In fact:

$$P_{X_0,X_1,\ldots,X_K}(a,b,c,\ldots,z) = P_{X_0}(a)\cdot P_{X_1|X_0}(b|a)\cdot P_{X_2|X_1}(c|b)\cdots P_{X_K|X_{K_1}}(z|y).$$

Homogeneous/Time-Invariant Markov Chains

Definition (Homogeneous Markov Chains)

A Markov chain $\{X_k\}_{k\geq 0}$ over an alphabet $\mathcal{X}$ is called *homogeneous* or *time-invariant* if the transition probability $P_{X_k|X_{k-1}}$ does not depend on the index k. That means, there exists a conditional probability mass function $W(\cdot|\cdot)$ such that:

$$P_{X_k|X_{k-1}}(a|b) = W(a|b), \quad \forall k = 1, 2, \ldots, \text{ and } a, b \in \mathcal{X}.$$

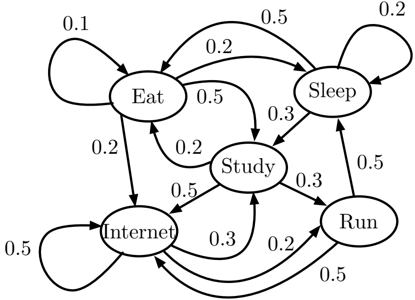
- The alphabet $\mathcal{X}$ is typically called the *state space* and W the *transition law* of the homogeneous Markov chain.

State-Transition Diagramme for Homogeneous Markov Chains

- A node for all possible states $a \in \mathcal{X}$ and an arrow from state b to state a labelled by the probability $W(a|b) > 0$. (If $W(a|b) = 0$ there is no arrow.)

- Each outgoing edge from state b represents a probability $W(\cdot|b)$
  $\Rightarrow$ the labels of all outgoing edges from a given node have to sum to 1!

Life in Lockdown:



Describing a Homogeneous Markov Chain with its Transition Matrix

- Transition matrix W: each row and each column is associated with a state $\rightarrow$ W is square of dimension $|\mathcal{X}| \times |\mathcal{X}|$

$$\mathrm{W} = \begin{pmatrix} W(a|a) & W(b|a) & W(c|a) & \cdots & W(z|a) \\ W(a|b) & W(b|b) & W(c|b) & \cdots & W(z|b) \\ \vdots & \dots & \ddots & \dots & \\ W(a|z) & \underbrace{W(b|z)}_{\mathrm{W}_{:,b}} & \cdots & \cdots & W(z|z) \end{pmatrix}$$

- Each row of W sums to 1 $\rightarrow$ a *(right) stochastic matrix*
- For any state b:

$$P_{X_1}(b) = \sum_{x \in \mathcal{X}} P_{X_0}(x) W(b|x) = \boldsymbol{\pi}_0 \cdot \mathrm{W}_{:,b}$$

  where $\boldsymbol{\pi}_k = (P_{X_k}(a), P_{X_k}(b), \dots, P_{X_k}(z))$.

- Summary for all $b \in \mathcal{X}$:

$$\boldsymbol{\pi}_1 = \boldsymbol{\pi}_0 \mathrm{W}.$$

The Markov Process in Matrix Notation

- Let $\boldsymbol{\pi}_k = (P_{X_k}(a), P_{X_k}(b), \ldots, P_{X_k}(z))$. Then:

$$\begin{aligned} \boldsymbol{\pi}_1 &= \boldsymbol{\pi}_0 \cdot \mathsf{W} \\ \boldsymbol{\pi}_2 &= \boldsymbol{\pi}_1 \cdot \mathsf{W} = \boldsymbol{\pi}_0 \cdot \mathsf{W} \cdot \mathsf{W} \\ &\vdots \\ \boldsymbol{\pi}_k &= \boldsymbol{\pi}_0 \cdot \mathsf{W}^k. \end{aligned}$$

$\rightarrow$ the statistics is determined by $\boldsymbol{\pi}_0$ and W

Transient and Recurrent States

Definition (Recurrent State Class)

Consider a homogeneous Markov process. A class of states $\mathcal{S} \subseteq \mathcal{X}$ is called *recurrent*, if the following two conditions hold:

1. For any two states $a, b \in \mathcal{S}$ there are positive integers k, i, j such that

$$\Pr[X_{k+i} = b | X_k = a] > 0 \qquad \text{and} \qquad \Pr[X_{k+j} = a | X_k = b] > 0.$$

   (We say that states a and b communicate.)

2. For any states $a \in \mathcal{S}$ and $b \in \mathcal{X} \backslash \mathcal{S}$ and for all $k, i > 0$:

$$\Pr[X_{k+i} = b | X_k = a] > 0.$$

If $\mathcal{X}$ is a recurrent class, the Markov process $\{X_k\}_{k \geq 0}$ is said *irreducible*.

Definition (Recurrent and Transient States)

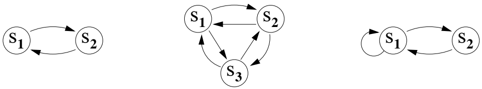
A state $a \in \mathcal{X}$ that belongs to some recurrent class is called *recurrent*. A state that does not belong to any recurrent class is called *transient*. For any transient state a:

$$\lim_{i \to \infty} \Pr[X_{k+i} = a | X_k = a] = 0$$

Periodicity of States And Aperiodic Chains

Definition (Periods of a states)

The period $d(x)$ of a state x is the smallest positive integer such that irrespective of the starting distribution $\Pr[X_{\ell+k} = x | X_k = x] = 0$ if ℓ is not a multiple of $d(x)$.



period of states:

Definition (Aperiodic Markov Chains)

A Markov chain $\{X_k\}$ is said aperiodic if $d(x) = 1$ for all states $x \in \mathcal{X}$.

A Stationary Process

Definition (Stationary Process)

A stochastic process $\{X_k\}_{k\geq 0}$ is called *stationary*, if for all integers $k, n \geq 0$:

$$P_{X_k, X_{k+1}, \ldots, X_{k+n}}(a, b, \ldots, z) = P_{X_0, X_1, \ldots, X_n}(a, b, \ldots, z), \quad \forall a, b, \ldots, z \in \mathcal{X}.$$

Theorem

A Markov process $\{X_k\}_{k\geq 0}$ *with transition matrix* W *and initial distribution* $\boldsymbol{\pi}_0$ *is stationary if, and only if,*

$$\boldsymbol{\pi}_0 = \boldsymbol{\pi}_0 \cdot \mathsf{W}.$$

Proof: The "only if" direction is trivial because $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_0 \cdot \mathsf{W}$.
To see the "if"-direction, notice that for any $k \geq 1$:

$$\boldsymbol{\pi}_k = \boldsymbol{\pi}_0 \cdot \mathsf{W}^k = \underbrace{\boldsymbol{\pi}_0 \cdot \mathsf{W}}_{=\boldsymbol{\pi}_0} \cdot \mathsf{W}^{k-1} = \boldsymbol{\pi}_0 \cdot \mathsf{W}^{k-1} = \underbrace{\boldsymbol{\pi}_0 \cdot \mathsf{W}}_{=\boldsymbol{\pi}_0} \cdot \mathsf{W}^{k-2} = \cdots = \boldsymbol{\pi}_0 \cdot \mathsf{W} = \boldsymbol{\pi}_0$$

and thus by Bayes' rule and the Markov property:

$$\begin{aligned}
&P_{X_k, X_{k+1}, \ldots, X_{k+n}}(a, b, \ldots, z) = P_{X_k}(a) P_{X_{k+1}|X_k}(b|a) \cdots P_{X_{k+n}|X_{k+n-1}}(z|y) \\
&= \boldsymbol{\pi}_0(a) \cdot W(b|a) \cdot W(c|b) \cdots W(z|y) = P_{X_0}(a) P_{X_1|X_0}(b|a) \cdots P_{X_n|X_{n-1}}(z|y) \\
&= P_{X_0, X_1, \ldots, X_n}(a, b, \ldots, z)
\end{aligned}$$

$\blacksquare$

More on Stationary Distributions

Consider a Markov chain $\{X_k\}_{k\geq 0}$ with transition matrix W.

- Any distribution $\boldsymbol{\pi}$ satisfying the fix-point equation

$$\boldsymbol{\pi} = \boldsymbol{\pi} \cdot \mathsf{W}$$

  is called a *stationary distribution* of this Markov chain.

- Any such $\boldsymbol{\pi}$ is an eigenvector of W corresponding to eigenvalue 1.

- Aperiodic and irreducible Markov chains have a unique stationary distribution $\boldsymbol{\pi}^*$.

- Transient states have 0 probability under $\boldsymbol{\pi}^*$.

Convergence of the Transition Matrix

Theorem

The following limit exists

$$\mathsf{W}^* := \lim_{N\to\infty} \mathsf{W}^N,$$

and W^* *is a stochastic matrix.*
For an irreducibile and aperiodic Markov chain:

$$\mathsf{W}^* = \mathbf{1}^\mathsf{T}\boldsymbol{\pi}^*,$$

where $\boldsymbol{\pi}^*$ *is the unique stationary distribution.*

Proof.

Omitted. $\square$

Convergence to A Stationary Process

Theorem

If the Markov chain $\{X_k\}_{k\geq 0}$ is aperiodic and irreducible, then for any initial distribution $\boldsymbol{\pi}_0$:

$$\lim_{N\to\infty} \boldsymbol{\pi}_N \to \boldsymbol{\pi}^*,$$

where $\boldsymbol{\pi}^*$ is the only stationary distribution of the Markov chain.

Proof:

$$\lim_{N\to\infty} \boldsymbol{\pi}_N = \lim_{N\to\infty} (\boldsymbol{\pi}_0 \cdot \mathsf{W}^N) = \boldsymbol{\pi}_0 \cdot \lim_{N\to\infty} \mathsf{W}^N = \underbrace{\boldsymbol{\pi}_0 \cdot \mathbf{1}^{\mathsf{T}}}_{=1} \boldsymbol{\pi}^*.$$

$\blacksquare$

Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

Telecom Paris, 27 November 2020

Lecture 2 – Markov Decision Processes and Dynamic Programming over a Finite Horizon

A Discrete-Time Dynamic System Model

- State evolution

$$X_{k+1} = f_k(X_k, U_k, W_k), \qquad k = 0, 1, 2, \ldots,$$

- X_k is the time-k state over a state space $\mathcal{X}$
- U_k is the time-k (control) action over a space $\mathcal{U}$
- W_k the random disturbance

Markov Decision Process (MDP) —A Markov Chain with Actions

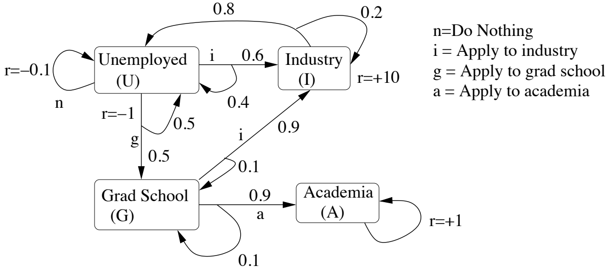
The discrete-time dynamic system is a *Markov decision process* if

- the sequence $\{W_k\}$ is memoryless; and
- a reward $R_u(x, x')$ is associated to each action u and pair of states $x, x' \in \mathcal{X}$

$\rightarrow$ Generalization of a Markov chain to incorporate actions and where the transition law depends on these actions:

$$P_{X_{k+1}|X_k,\ldots,X_0,U_k,\ldots,U_0}(a|b,\ldots,z,u,\ldots,v) = P_{X_{k+1}|X_k,U_k}(a|b,u), \quad \forall a,b,\ldots,z \in \mathcal{X}, u,v \in \mathcal{U}.$$

An MDP Example with Graph Representation



- Boxes are states; labels on arrows designate actions and transition probabilities. E.g.:

$$\Pr[X_{k+1} = \text{“I”} \,|\, X_k = \text{“U”}, U_k = \text{“i”}] = 0.6.$$

Finite-Horizon Dynamic Programming Problem Setup

(Slightly more general than introduced for MDPs)

- Discrete-time dynamic system:

$$X_{k+1} = f_k(X_k, U_k, W_k), \qquad k = 0, 1, 2, \ldots, N-1$$

where given (X_k, U_k) the noise W_k is conditionally independent of $(X_0, \ldots, X_{k-1}, U_1, \ldots, U_{k-1}, W_1, \ldots, W_{k-1})$

- N is called the *horizon* of the control problem

- Admissible control sets $\{\mathcal{U}_k(a)\}_{a \in \mathcal{X}}$ for action $U_k = \mu_k(X_k)$
$\rightarrow$ The set of functions $\mu_0, \ldots, \mu_{N-1}$ is called a *policy* π

- Additive expected cost

$$\mathbb{E}\left[g_N(X_N) + \sum_{k=0}^{N-1} g_k(X_k, U_k, W_k)\right] = \mathbb{E}_{\{W_k\}}\left[g_N(X_N) + \sum_{k=0}^{N-1} g_k(X_k, \mu_k(X_k), W_k)\right]$$

where $g_N(X_N)$ denotes a terminal cost

Decomposition of Expected Cost

- Expected time i-to-j cost starting from state $a \in \mathcal{X}$:

$$J_{i\to j,\pi}(a) = \mathbb{E}\left[\sum_{k=i}^{j} g_k(X_k, \mu_k(X_k), W_k) \middle| X_i = a\right], \quad 0 \le i < j \le N,$$

where $g_N(X_N, \mu_N(X_N), W_N) := g_N(X_N)$.

- Decomposition of finite-horizon expected cost for $i < j \le N$:

$$J_{i\to N,\pi}(a) = \mathbb{E}\left[g_N(X_N) + \sum_{k=i}^{N-1} g_k(X_k, \mu_k(X_k), W_k) \middle| X_j = b, X_i = a\right]$$

$$= \sum_{b\in\mathcal{X}} \Pr[X_j = b | X_i = a]\mathbb{E}\left[g_N(X_N) + \sum_{k=i}^{N-1} g_k(X_k, \mu_k(X_k), W_k) \middle| X_j = b, X_i = a\right]$$

$$= \sum_{b\in\mathcal{X}} \Pr[X_j = b | X_i = a]\mathbb{E}\left[g_N(X_N) + \sum_{k=j}^{N-1} g_k(X_k, \mu_k(X_k), W_k) \middle| X_j = b, X_i = a\right]$$

$$+ \sum_{b\in\mathcal{X}} \Pr[X_j = b | X_i = a]\mathbb{E}\left[\sum_{k=i}^{j-1} g_k(X_k, \mu_k(X_k), W_k) \middle| X_j = b, X_i = a\right]$$

$$= \sum_{b\in\mathcal{X}} \Pr[X_j = b | X_i = a] J_{j\to N,\pi}(b) + J_{i\to j-1,\pi}(a)$$

Minimizing the Expected Finite-Horizon Cost

- Minimize expected cost for $a \in \mathcal{X}$: $\quad J^*_{0\to N}(a) = \min_\pi J_{0\to N,\pi}(a)$

- Decomposition of optimization problem:

$$\min_\pi J_{0\to N,\pi}(a) = \min_{\mu_0}\left[J_{0\to 0,\mu_0}(a) + \min_{\mu_1,\dots,\mu_{N-1}} \sum_{b\in\mathcal{X}} \Pr[X_1 = b|X_0 = a] J_{1\to N,\pi}(b)\right]$$

$$\geq \min_{\mu_0}\left[J_{0\to 0,\mu_0}(a) + \sum_{b\in\mathcal{X}} \Pr[X_1 = b|X_0 = a] \min_{\mu_{b,1},\dots,\mu_{b,N-1}} J_{1\to N,\pi_b}(b)\right]$$

where equality holds when optimal policies $\mu_{b,1},\dots,\mu_{b,N-1}$ don't depend on b.

$$\min_\pi J_{1\to N,\pi}(b) \geq \min_{\mu_1}\left[J_{1\to 1,\mu_1}(b) + \sum_{c\in\mathcal{X}} \Pr[X_2 = c|X_1 = b] \min_{\mu_{c,2},\dots,\mu_{c,N-1}} J_{2\to N,\pi_c}(c)\right]$$

$$\vdots$$

$$\min_\pi J_{N-1\to N,\pi}(x) \geq \min_{\mu_{N-2}}\Big[J_{N-1\to N-1,\mu_{N-1}}(x) + \sum_{y\in\mathcal{X}} \Pr[X_N = y|X_{N-1} = x] \min_{\mu_{y,N-1}} J_{N\to N,\pi_y}(y)\Big]$$

- Will see: optimal $\mu_{a,i},\dots,\mu_{a,N-1}$ don't depend on $a \Rightarrow$ Ineq. are equalities
- Find the optimal solution starting backwards!!

Optimal Dynamic Programming Algorithm

- For each $x_N \in \mathcal{X}$ initialize $J^*_{N\to N}(x_N) = g_N(x_N)$
  $\to$ trivially the same μ_N achieves optimal $J^*_{N\to N}(x_N)$ for all $x_N \in \mathcal{X}$

- For each $i = N-1, \dots, 0$ calculcate for each $x_i \in \mathcal{X}$:

$$J^*_{i\to N}(x_i)$$

$$:= \min_{\mu_i} \left[J_{i\to i,\mu_i}(x_i) + \sum_{x_{i+1}\in\mathcal{X}} \Pr[X_{i+1} = x_{i+1} | X_i = x_i] J^*_{i+1\to N}(x_{i+1}) \right]$$

$$= \min_{\mu_i} \left[\mathbb{E}_{W_i} \left[g_i(x_i, \mu_i(x_i), W_i) + J^*_{i+1\to N}(X_{i+1}) \middle| X_i = x_i \right] \right]$$

$\to$ If optimal policies $\mu^*_{i+1}, \dots, \mu^*_N$ for $J^*_{i+1\to N}(x_{i+1})$ don't depend on $x_{i+1} \in \mathcal{X}$, then optimal policies $\mu^*_i, \mu_{i+1}, \dots, \mu^*_N$ for $J^*_{i\to N}(x_i)$ don't depend on x_i!

Optimality Principle for Finite-Horizon Dynamic Programming

Theorem (Optimality Principle)

Let $\pi^* = (\mu_0^*, \mu_1^*, \mu_2^*, \ldots, \mu_{N-1}^*)$ *be an optimal policy for* $J_{0\to N,\pi}$*:*

$$J_{0\to N,\pi^*}(a) = \min_{\pi} J_{0\to N,\pi}(a) =: J^*_{0\to N}(a), \qquad \forall a \in \mathcal{X}.$$

Then $\forall b \in \mathcal{X}$ *the truncated policy* $\pi^*_{i\to N} := (\mu_i^*, \ldots, \mu_{N-1}^*)$ *minimizes the sub-problem* $J_{i\to N,\pi}$*:*

$$J_{i\to N,\pi^*_{i\to N}}(b) = \min_{\pi} J_{i\to N,\pi}(b) =: J^*_{i\to N}(b), \qquad \forall b \in \mathcal{X}.$$

Proof by Contradiction: Given policy $\pi_{i\to N} = (\mu_0, \mu_1, \ldots, \mu_{N-1})$ satisfying

$$J_{i\to N,\pi}(b) < J_{i\to N,\pi^*}(b), \quad \forall b \in \mathcal{X}.$$

Then for all $a \in \mathcal{X}$ and policy $\tilde{\pi} = (\mu_0^*, \mu_1^*, \ldots, \mu_{i-1}^*, \mu_i, \ldots, \mu_{N-1})$:

$$\begin{aligned}
J_{0\to N,\pi^*}(a) &= \sum_{b\in\mathcal{X}} \Pr[X_i = b | X_0 = a] J_{i\to N,\pi^*}(b) + J_{0\to i-1,\pi^*}(a) \\
&> \sum_{b\in\mathcal{X}} \Pr[X_i = b | X_0 = a] J_{i\to N,\pi}(b) + J_{0\to i-1,\pi^*}(a) \\
&= J_{0\to N,\tilde{\pi}}(a)
\end{aligned}$$

Example: Inventory Control

- state x_k: stock at the beginning of period k
- action u_k: stock order (and delivery) at the beginning of period k
- disturbance w_k: random demand during period k
- state evolution:

$$x_{k+1} = f(x_k, u_k, w_k) = x_k + u_k - w_k.$$

- cost $g_k(x_k, u_k, w_k)$ in period k consists of inventory cost/penalty $r(x_k)$ and purchase cost cu_k:

$$g_k(x_k, u_k, w_k) = r(x_k) + c \cdot u_k$$

- Wish to minimize total expected cost over horizon N:

$$J_{0\to N,\pi} = \mathbb{E}\left[\sum_{k=0}^{N} r(x_k) + \sum_{k=0}^{N-1} c \cdot u_k \middle| X_0 = a \right], \quad a \geq 0.$$

Optimal DP Algorithm for the Inventory Control Example

- Initialize $J^*_{N\to N}(x_N) = r(x_N)$
- First iteration:

$$\begin{aligned} J^*_{N-1\to N}(x_{N-1}) &= \min_{u_{N-1}} \{r(x_{N-1}) + cu_{N-1} + \mathbb{E}[r(X_N)]\} \\ &= r(x_{N-1}) + \min_{u_{N-1}} \left\{cu_{N-1} + \mathbb{E}_{W_{N-1}}[r(x_{N-1} + u_{N-1} + W_{N-1})]\right\} \end{aligned}$$

- Second iteration:

$$\begin{aligned} J^*_{N-2\to N} &= \min_{u_{N-2}} \{r(x_{N-2}) + cu_{N-2} + \mathbb{E}[J^*_{N-1\to N}(X_{N-1})]\} \\ &= r(x_{N-2}) + \min_{u_{N-2}} \left\{cu_{N-2} + \mathbb{E}_{W_{N-2}}[J^*_{N-1\to N}(x_{N-2} + u_{N-2} + W_{N-2})]\right\} \end{aligned}$$

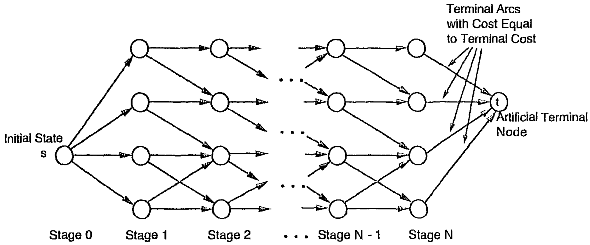
- i-th iteration:

$$J^*_{N-i\to N} = r(x_{N-i}) + \min_{u_{N-i}} \left\{cu_{N-i} + \mathbb{E}_{W_{N-i}}[J^*_{N-i-1\to N}(x_{N-i} + u_{N-i} + W_{N-i})]\right\}$$

- Solution obtained after N iterations: $J^*_{0\to N}$

Deterministic MDPs and Shortest-Path Problems

- No disturbance → state evolution $x_{k+1} = f(x_k, u_k)$ and cost $g_k(x_k, u_k)$
- Graph representation:

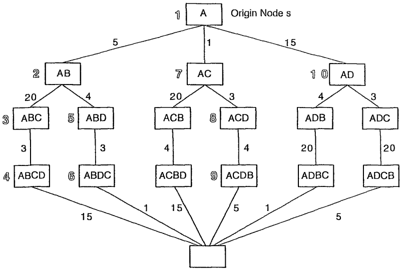


  - At each stage $k = 1, 2, \ldots, N$ there is a node for each $x_k \in \mathcal{X}$
  - Arrows indicate transitions for different actions → label arrows with actions u_k and costs $g_k(x_k, u_k)$

- Total cost $J_{0\to N,\pi}$ is the sum of the costs on the path indicated by π

Finding minimum total cost $J_{0\to N,\pi}$ equivalent to finding "shortest path"
→ DP algorithm can be run in reverse order

Travelling Salesman Problem and Label Correcting Method



	5	1	15
5		20	4
1	20		3
15	4	3	

- State space depends on stage k

Initialize $d_1 = 0$ and
$d_2 = \cdots = d_t = \infty$

Label Correcting Algorithm

Step 1: Remove a node i from OPEN and for each child j of i, execute step 2.

Step 2: If $d_i + a_{ij} < \min\{d_j, \text{UPPER}\}$, set $d_j = d_i + a_{ij}$ and set i to be the parent of j. In addition, if $j \neq t$, place j in OPEN if it is not already in OPEN, while if $j = t$, set UPPER to the new value $d_i + a_{it}$ of d_t.

Step 3: If OPEN is empty, terminate; else go to step 1.

Iter. No.	Node Exiting OPEN	OPEN at the End of Iteration	UPPER
0	-	1	∞
1	1	2, 7,10	∞
2	2	3, 5, 7, 10	∞
3	3	4, 5, 7, 10	∞
4	4	5, 7, 10	43
5	5	6, 7, 10	43
6	6	7, 10	13
7	7	8, 10	13
8	8	9, 10	13
9	9	10	13
10	10	Empty	13

- Dijkstra's method always chooses the node in OPEN with smallest d_i.

Dynamic Programming in a Hidden Markov Model

- In a *Hidden Markov Model (HMM)* or *Partially Observable Markov Process (POMP)*, an observer does not observe the state sequences $X_0, X_1, \ldots, X_N$ directly but a related sequence $Z_1, \ldots, Z_N$, where

$$P_{X_0,X_1,\ldots,X_N,Z_1,\ldots,Z_N} = P_{X_0} \cdot \prod_{k=1}^{N} P_{X_k|X_{k-1}} \cdot P_{Z_k|X_k,X_{k-1}}.$$

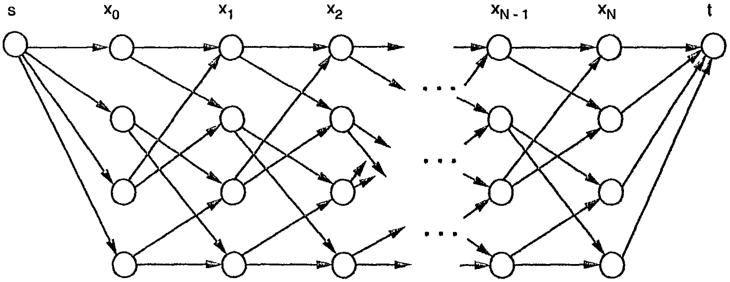
- Observe $z_1, \ldots, z_N$ and solve

$$\min_{x_0,x_1,\ldots,x_N} -\log P_{X_0,X_1,\ldots,X_N,Z_1,\ldots,Z_N}(x_0, x_1, \ldots, x_N, z_1, \ldots, z_N)$$
$$= \min_{x_0,x_1,\ldots,x_N} \left[-\log P_{X_0}(x_0) - \sum_{k=1}^{N} \log P_{X_k|X_{k-1}}(x_k|x_{k-1}) P_{Z_k|X_k,X_{k-1}}(z_k|x_k, x_{k-1}) \right]$$

$\rightarrow$ Apply Forward DP algorithm on a Trellis

The Viterbi Algorithm

- Trellis:



Edges from s to x_0 are labeled with P_{X_0}, edges from x_N to t by 0 and edges from x_{k-1} to x_k by $-\log P_{X_k|X_{k-1}}(x_k|x_{k-1})P_{Z_k|X_k,X_{k-1}}(z_k|x_k,x_{k-1})$

- Shortest Path from s to t solves minimization problem

- Apply forward DP algorithm and cut the branches that are suboptimal

Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

Telecom Paris, 8 December 2020

Lecture 3 – Dynamic Programming over an Infinite Horizon: The Discounted Case

Review of Lecture 2: Finite Horizon and Decomposition of the Cost

- Discrete-time dynamic system:

$$X_{k+1} = f_k(X_k, \mu_k(X_k), W_k), \qquad k = 0, 1, 2, \ldots, N-1$$

  $\{W_k\}$ is independent and identically distributed (i.i.d.)

- Minimize total cost for given initial state $a \in \mathcal{X}$:

$$J^*_{0\to N}(a) := \min_\pi \underbrace{\mathbb{E}\left[\sum_{k=0}^{N-1} g_k(X_k, \mu_k(X_k), W_k) + g_N(X_N) \middle| X_0 = a \right]}_{=:J_{0\to N,\pi}(a)}$$

- Optimal *Backward DP Algorithm*: Initialize $J^*_{N\to N}(x_N) := g_N(x_N)$ and compute for $i = N-1, \ldots, 0$

$$J^*_{i\to N}(x_i) = \min_{\mu_i} \left(\mathbb{E}\Big[g_i(x_i, \mu_i(x_i), W_i) + \sum_{x_{i+1}\in\mathcal{X}} \Pr[X_{i+1} = x_{i+1} | X_i = x_i] J^*_{i+1\to N}(x_{i+1}) \right)$$

$$= \min_{\mu_i} \mathbb{E}_{W_i} \Big[g_i(x_i, \mu_i(x_i), W_i) + J^*_{i+1\to N}\Big(f_i(x_i, \mu_i(x_i), W_i) \Big) \Big]$$

- For deterministic problems optimal DP algorithm can be run forwards

Optimality of Memoryless Policies

- Restriction to memoryless policies $u_i = \mu_i(x_i)$ is without loss of optimality. (I.e., there is no need to consider policies of the form $u_i = \mu_i(x_0, \ldots, x_i, u_1, \ldots, u_{i-1})$.)

- Recall

$$J^*_{i\to N}(x_i) = \min_{\mu_i} \left(\mathbb{E}\left[g_i(x_i, \mu_i(x_i), W_i) + \sum_{x_{i+1}\in\mathcal{X}} \Pr[X_{i+1} = x_{i+1}|X_i = x_i] J^*_{i+1\to N}(x_{i+1}) \right) \right.$$

- $J^*_{i\to N}(x_i)$ only depends on $P_{X_{i+1}|X_i}$ and $P_{X_i U_i}$ $\to$ introducing memory would have no effect at all on the value of $J^*_{i\to N}(x_i)$.

- Deterministic policies suffice because the minimum has a deterministic solution

Infinite-Horizon Dynamic Programming with Discounted Costs

- *Time-invariant* discrete-time dynamic system:

$$X_{k+1} = f(X_k, U_k, W_k), \qquad k = 0, 1, 2, \ldots,$$

- Bounded time-invariant cost function $g(x, u, w) \in [-M, M]$

Definition (Optimal Discounted Cost)

Given a *discounting factor* $\gamma > 0$, the *discounted expected cost* for policy $\pi = (\mu_0, \mu_1, \ldots,)$ is:

$$J_\pi(a) := \mathbb{E}_{\{W_k\}}\left[\sum_{k=0}^{\infty} \gamma^k g(X_k, \mu_k(X_k), W_k) \middle| X_0 = a \right]$$

The *optimal infinite-horizon discounted cost* is $J^*(a) := \min_\pi J_\pi(a)$

A Closer Look at the Finite-Horizon Discounted Cost Problem

- The finite-horizon cost for our problem and policy π. $\forall L < N$:

$$
\begin{aligned}
& J_{0\to N,\pi}(a) \\
&= \mathbb{E}_{|X_0=a}\left[\sum_{k=0}^{L-1}\gamma^k g(X_k,\mu_k(X_k),W_k) + \sum_{k=L}^{N-1}\gamma^k g(X_k,\mu_k(X_k),W_k) + \gamma^N g_N(X_N)\right] \\
&\le \mathbb{E}_{|X_0=a}\left[\sum_{k=0}^{L-1}\gamma^k g(X_k,\mu_k(X_k),W_k)\right] + \gamma^L g_L(X_L) + \sum_{k=L}^{N}\gamma^k M - \gamma^L g_L(X_L) \\
&\le J_{0\to L,\pi}(a) + M\gamma^L\left(1 + \frac{1-\gamma^{N-L+1}}{1-\gamma}\right)
\end{aligned}
$$

- Let $N \to \infty$ and take $\min_\pi$ on both sides:

$$J^*(a) := \min_\pi \lim_{N\to\infty} J_{0\to N,\pi}(a) \le \min_\pi J_{0\to L,\pi}(a) + M\gamma^L \frac{2-\gamma}{1-\gamma}$$

Similarly, we obtain

$$J^*(a) \ge J^*_{0\to L}(a) - M\gamma^L \frac{2-\gamma}{1-\gamma}$$

Optimal Infinite-Horizon Discounted Cost as a Limit

By a sandwiching argument and $L \to \infty$:

Theorem

The Optimal Infinite-Horizon Discounted Cost can be obtained as:

$$J^*(a) = \lim_{L \to \infty} J^*_{0 \to L}(a), \quad \forall a \in \mathcal{X},$$

irrespective of the termination costs $\{\gamma^L g_L(X_L)\}$.

- Is there a way to efficiently compute this limit?
  $\rightarrow$ Yes, because of time-invariance and since the starting point does not matter!

Rephrasing the Finite-Horizon Cost

- Finite-horizon Optimal DP algorithm:

$$J^*_{i\to N}(a) := \min_{\mu} \mathbb{E}_{W_i}\left[\gamma^i g(a,\mu(a),W_i) + J^*_{i+1\to N}(f(a,\mu(a),W_i))\right],$$

  for starting condition $J^*_{N\to N}(a) := \gamma^N g_N(a)$ for all $a \in \mathcal{X}$.

- For $i < N$ define $V_{N-i}(a) := \frac{1}{\gamma^i} J^*_{i\to N}(a)$ and $W'_{N-i} := W_i$, and $k = N - i$:

$$V_0(a) = J^*_{N\to N}(a)$$
$$V_k(a) = \min_{\mu} \mathbb{E}_{W'_k}\left[g(a,\mu(a),W'_k) + \gamma V_{k-1}(f(a,\mu(a),W'_k))\right], \quad k = 1,\ldots,N$$

- Recursion independent of N and $\forall N$: $V_N(a) = J^*_{0\to N}(a)$! (with same g_N.)

Lemma

$$J^*(a) = \lim_{N\to\infty} V_N(a),$$

where

$$V_k = \min_{\mu} \mathbb{E}[g + \gamma V_{k-1}], \quad k = 1,2,\ldots,$$

and starting vector V_0 can be arbitrary.

The Value-Iteration Algorithm for Dynamic Programming

- Finds an approximation to the solution vector J^* for an infinite-horizon DP problem with discounted and bounded costs

- Algorithm:
  - Select an arbitrary starting vector $V_0 \in \mathbb{R}^{|\mathcal{X}|}$
  - For $k = 1, 2, \ldots,$ calculate for each $a \in \mathcal{X}$:
  $$V_k(a) = \min_{\mu} \mathbb{E}_W[g(a, \mu(a), W) + \gamma V_{k-1}(f(a, \mu(a), W)].$$
  - Stop according to some convergence criterion, for example when the value on each component does not change more than a given value ϵ.

- How fast does it converge? Error bounds?

- Attention: In the literature V is often also called J

Exponential Decay on Difference of Iterations

Lemma

Given two bounded initial vectors V_0 and V_0' such that

$$\max_{a\in\mathcal{X}} |V_0(a) - V_0'(a)| \le c.$$

If $V_1, \dots, V_k$ and $V_1', \dots, V_k'$ are obtained from the DP recursion for V_0 and V_0', respectively:

$$\max_{a\in\mathcal{X}} |V_k(a) - V_k'(a)| \le \alpha^k \max_{a\in\mathcal{X}} |V_0(a) - V_0'(a)|.$$

Proof: By induction:

$$\begin{aligned}
V_1(a) &= \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V_0(f(a,\mu(a),W))] \\
&\le \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V_0'(f(a,\mu(a),W))] + \gamma c = V_1'(a) + \gamma c \\
V_k(a) &= \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V_{k-1}(f(a,\mu(a),W))] \\
&\le \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V_{k-1}'(f(a,\mu(a),W))] + \gamma\gamma^{k-1} c = V_k'(a) + \gamma^k c
\end{aligned}$$

Similarly, $V_1(a) \ge V_1'(a) - \gamma c$ and $V_k(a) \ge V_k'(a) - \gamma^k c$

Error Bounds on the Value-Iteration Algorithm

- By Bellman's equation ahead, $V_0' = J^*$ implies $V_1' = \cdots V_k' = J^*$ and thus

$$\max_{a \in \mathcal{X}} |V_k(a) - J^*(a)| \leq \alpha^k \max_{a \in \mathcal{X}} |V_0(a) - J^*(a)|.$$

- The error in the value-iteration algorithm vanishes exponentially fast with each iteration

The Operator Interpretation

- Operator $\mathbb{T}$ (or $\mathbb{T}_{f,g,\gamma}$) acts on vector $V \in \mathcal{R}^{|\mathcal{X}|}$ componentwise as:

$$(\mathbb{T}V)(a) = \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V(f(a,\mu(a),W))], \quad \forall a \in \mathcal{X}.$$

- Optimal DP iteration is described as: $V_{k+1} = \mathbb{T}V_k$.

- The operator $\mathbb{T}$ is *contracting* since $\exists \rho \in (0,1)$:

$$\|\mathbb{T}(J) - \mathbb{T}(J')\| \leq \rho \|J - J'\|, \quad \forall J, J',$$

  where here $\|\cdot\|$ denotes the infinity norm (i.e., the maximum component)

- Irrespective of V, as $k \to \infty$ the operator $\mathbb{T}^k V = \underbrace{\mathbb{T}(\mathbb{T}(\cdots \mathbb{T}}_{k \text{ applications of } \mathbb{T}}(V)))$ converges to a unique J^* that satisfies the *fix-point equation*

$$J^* = \mathbb{T}J^*$$

Bellman's Equation

Theorem

The cost vector J^* is optimal if, and only if, it satisfies

$$J^*(a) = \min_{\mu} \mathbb{E}_W[g(a, \mu(a), W) + \gamma J^*(f(a, \mu(a), W))], \quad \forall a \in \mathcal{X}.$$

There is a unique finite cost-vector J^* satisfying above equation.

Proof: "If"-direction: Set J^* as starting vector in iteration.
"Only if"-direction uses the previous bounds. $\forall a \in \mathcal{X}$:

$$\begin{aligned} J^*(a) - M\gamma^{L+1}\frac{2-\gamma}{1-\gamma} &\leq V_{L+1} = \min_{\mu} \mathbb{E}_W[g(a, \mu(a), W) + \gamma V_L(f(a, \mu(a), W))] \\ &\leq \min_{\mu} \mathbb{E}_W[g(a, \mu(a), W) + \gamma J^*(f(a, \mu(a), W))] + M\gamma^L \frac{2-\gamma}{1-\gamma}. \end{aligned}$$

Similarly:

$$J^*(a) + M\gamma^{L+1}\frac{2-\gamma}{1-\gamma} \geq \min_{\mu} \mathbb{E}_W[g(a, \mu(a), W) + \gamma J^*(f(a, \mu(a), W))] - M\gamma^L \frac{2-\gamma}{1-\gamma}.$$

Taking $L \to \infty$ by sandwiching argument proves "only-if" direction.
Uniqueness follows by convergence of $\{V_k\}_{k \geq 0}$ irrespective of V_0.

About Stationary Policies

- A policy of the form $\pi = (\mu, \mu, \mu, \ldots)$ is called stationary.
- For any stationary policy μ and arbitrary initial vector V_0:

$$V_{k,\mu}(a) = \mathbb{E}_W[g(a, \mu(a), W) + \gamma V_{k-1,\mu}(f(a, \mu(a), W))]$$

converges for each $a \in \mathcal{X}$. Call the convergence point $J_\mu(a)$.

- If $V_{1,\mu}(a) \leq V_{0,\mu}(a)$ for all $a \in \mathcal{X}$, then $V_{k,\mu}$ is a decreasing sequence

Lemma (Optimality of Stationary Policies)

A stationary policy μ^* is optimal if, and only if,

$$\begin{aligned}&\mathbb{E}_W[g(a, \mu^*(a), W) + \gamma J^*(f(a, \mu^*(a), W))] \\ &\quad = \min_\mu \mathbb{E}_W[g(a, \mu(a), W) + \gamma J^*(f(a, \mu(a), W))], \quad \forall a \in \mathcal{X}.\end{aligned}$$

Proof: Follows essentially from Bellman's equation and the uniqueness of the solution J^*.

Finding an Improved Stationary Policy

Theorem

Let μ and $\bar{\mu}$ be stationary policies satisfying $\forall a \in \mathcal{X}$:

$$\mathbb{E}_W[g(a, \bar{\mu}(a), W) + \gamma J_\mu(a, \bar{\mu}(a), W)] = \min_u \mathbb{E}_W\left[g(a, u) + \gamma J_\mu(f(a, u, W))\right].$$

Then,

$$J_{\bar{\mu}}(a) \leq J_\mu(a), \quad \forall a \in \mathcal{X},$$

where inequality is strict for at least one $a \in \mathcal{X}$ whenever μ is not optimal.

Proof:

$$\begin{aligned}
\underbrace{J_\mu(a)}_{V_{0,\bar{\mu}}} &= \mathbb{E}[g(a, \mu(a), W) + \gamma J_\mu(a)(f(a, \mu(a), W))] \\
&\geq \underbrace{\mathbb{E}[g(a, \bar{\mu}(a), W) + \gamma J_\mu(a)(f(a, \bar{\mu}(a), W))]}_{V_{1,\bar{\mu}}} \\
&\geq V_{2,\bar{\mu}} \geq V_{3,\bar{\mu}} \geq \ldots \\
&\geq J_{\bar{\mu}}(a).
\end{aligned}$$

Policy Iteration Algorithm

- Finds the *exact* solution vector J^* for an infinite-horizon DP problem with discounted and bounded costs
- Algorithm:
  - Select an arbitrary policy μ_0 and find J_{μ_0} by solving the linear system of equations:

$$J_{\mu_0}(a) = \mathbb{E}[g(a, \mu_0(a), W)] + \gamma \mathbb{E}[J_{\mu_0}(f(a, \mu_0(a), W))], \quad a \in \mathcal{X}.$$

  - For $k = 1, 2, \ldots$ solve the minimization problem

$$\mu_k(a) := \text{argmin}_{u \in \mathcal{U}} \mathbb{E}_W[g(a, u, W) + \gamma J_{\mu_{k-1}}(f(a, u, W)], \quad a \in \mathcal{X}.$$

  and find J_{μ_k} by solving the linear system of equations:

$$J_{\mu_k}(a) = \mathbb{E}[g(a, \mu_k(a), W)] + \gamma \mathbb{E}[J_{\mu_k}(f(a, \mu_k(a), W))], \quad a \in \mathcal{X}.$$

  - Stop when $\mu_k = \mu_{k-1}$ and produce $J^* = J_{\mu_{k-1}}$
- Advantage: There is only a finite number of stationary policies and thus the algorithm finds the exact optimal discounted cost J^*.

A Simple Binary Example

- Let $\mathcal{X} = \{a, b\}$ and $\mathcal{U} = \{1, 2\}$. Moreover, $W_i \sim \mathcal{B}(1/4)$ and $\gamma = 0.9$.
- Transition function: $f(x, u, w) = a$ if $(u = 1, w = 1)$ or $(u_2 = 2, w = 0)$, and $f(x, u, w) = b$ else
- Cost function: $\mathbb{E}_W[g(a, 1, W)] = 2, \quad \mathbb{E}_W[g(a, 2, W)] = 0.5,$ $\mathbb{E}_W[g(b, 1, W)] = 1, \quad \mathbb{E}_W[g(b, 2, W)] = 3.$
- Value iteration algorithm with starting point $V_0 = (0, 0)^\mathsf{T}$:

$$
\begin{aligned}
V_1(a) &= \min_{\mu} \left(\mathbb{E}[g(a, \mu(a), W)] + \mathbb{E}[\gamma V_0(f(a, \mu(a), W))]\right) \\
&= \min_{u \in \{1,2\}} \mathbb{E}[g(a, u, W)] = \min\{2, 0.5\} = 0.5. \\
V_1(b) &= \min_{u \in \{1,2\}} \mathbb{E}[g(a, u, W)] = \min\{1, 3\} = 1.
\end{aligned}
$$

$$
\begin{aligned}
V_2(a) &= \min\left\{\mathbb{E}[g(a, 1, W) + \gamma V_1(f(a, 1, W))], \mathbb{E}[g(a, 2, W) + \gamma V_1(f(a, 2, W))]\right\} \\
&= \min\{2 + 0.9 \cdot (0.5 \cdot 3/4 + 1 \cdot 1/4), 0.5 + 0.9 \cdot (0.5 \cdot 1/4 + 1 \cdot 3/4)\} \\
&= \min\{2 + 0.9 \cdot 5/8, 0.5 + 0.9 \cdot 7/8\} = 0.5 + 0.9 \cdot 7/8 = 1.2875 \\
V_2(b) &= \min\{1 + 0.9 \cdot 5/8, 3 + 0.9 \cdot 7/8\} = 1 + 0.9 \cdot 5/8 = 1.5625
\end{aligned}
$$

Example Continued

- Value iteration algorithm continued:

$$
\begin{aligned}
V_3(a) &= 1.844 & \qquad V_3(b) &= 2.220 \\
V_4(a) &= 2.414 & V_4(b) &= 2.745 \\
V_5(a) &= 2.896 & V_5(b) &= 3.247 \\
&\vdots & &\vdots \\
V_{15}(a) &= 5.783 & V_{15}(b) &= 6.128
\end{aligned}
$$

- Policy iteration algorithm with initial policy $\mu_0(a) = 1$ and $\mu_0(b) = 2$:
  - Policy evaluation to determine J_{μ_0}:

$$
J_{\mu_0}(a) = 2 + 0.9 \cdot (J_{\mu_0}(a) \cdot 3/4 + J_{\mu_0}(b) \cdot 1/4)
$$
$$
J_{\mu_0}(b) = 3 + 0.9 \cdot (J_{\mu_0}(a) \cdot 1/4 + J_{\mu_0}(b) \cdot 3/4)
$$

$$
\Rightarrow J_{\mu_0} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \underbrace{\begin{pmatrix} 0.9 \cdot 3/4 & 0.9 \cdot 1/4 \\ 0.9 \cdot 1/4 & 0.9 \cdot 3/4 \end{pmatrix}}_{\substack{\text{state transition matrix} \\ P_{\mu_0} \text{ from } X_0 \text{ to } X_1}} J_{\mu_0} = \begin{pmatrix} 24.091 \\ 25.909 \end{pmatrix}
$$

Example Continued II

- Policy improvement to determine μ_1:

$$\begin{aligned}\mu_1(a) &= 1 + \mathbb{1}\big\{\mathbb{E}_W[g(a,1,W) + \gamma J_{\mu_0}(f(a,1,W))] \\ &\qquad\qquad > \mathbb{E}_W[g(a,2,W) + \gamma J_{\mu_0}(f(a,2,W))]\big\} \\ &= 1 + \mathbb{1}\big\{2 + 0.9 \cdot 3/4 \cdot 24.091 + 0.9 \cdot 1/4 \cdot 25.909 \\ &\qquad\qquad > 0.5 + 0.9 \cdot 1/4 \cdot 24.091 + 0.9 \cdot 3/4 \cdot 25.909\big\} \\ &= 1 + \mathbb{1}\big\{24.909 > 23.409\big\} = 2\end{aligned}$$

$$\begin{aligned}\mu_1(b) &= 1 + \mathbb{1}\big\{1 + 0.9 \cdot 3/4 \cdot 24.091 + 0.9 \cdot 1/4 \cdot 25.909 \\ &\qquad\qquad > 3 + 0.9 \cdot 1/4 \cdot 24.091 + 0.9 \cdot 3/4 \cdot 25.909\big\} \\ &= 1 + \mathbb{1}\big\{22.909 > 25.909\big\} = 1\end{aligned}$$

- Policy evaluation to determine J_{μ_1}:

$$J_{\mu_1}(a) = 0.5 + 0.9 \cdot (J_{\mu_1}(a) \cdot 1/4 + J_{\mu_1}(b) \cdot 3/4)$$
$$J_{\mu_1}(b) = 1 + 0.9 \cdot (J_{\mu_1}(a) \cdot 3/4 + J_{\mu_1}(b) \cdot 1/4)$$

$$\Rightarrow J_{\mu_1} = \underbrace{\begin{pmatrix} 0.5 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.9 \cdot 1/4 & 0.9 \cdot 3/4 \\ 0.9 \cdot 3/4 & 0.9 \cdot 1/4 \end{pmatrix}}_{\substack{\text{state transition matrix} \\ P_{\mu_1} \text{ from } X_1 \text{ to } X_2}} J_{\mu_1} = \begin{pmatrix} 7.3276 \\ 7.6724 \end{pmatrix}$$

Example Continued III

- Policy improvement to determine μ_2:

$$\begin{aligned}\mu_2(a) &= 1 + \mathbb{1}\big\{\mathbb{E}_W[g(a,1,W) + \gamma J_{\mu_1}(f(a,1,W)] \\ &\qquad\qquad > \mathbb{E}_W[g(a,2,W) + \gamma J_{\mu_1}(f(a,2,W)]\big\} \\ &= 1 + \mathbb{1}\big\{2 + 0.9 \cdot 3/4 \cdot 27.3276 + 0.9 \cdot 1/4 \cdot 7.6724 \\ &\qquad\qquad > 0.5 + 0.9 \cdot 1/4 \cdot 7.3276 + 0.9 \cdot 3/4 \cdot 7.6724\big\} \\ &= 1 + \mathbb{1}\big\{8,6724 > 7.3276\big\} = 2\end{aligned}$$

$$\begin{aligned}\mu_2(b) &= 1 + \mathbb{1}\big\{1 + 0.9 \cdot 3/4 \cdot 7.3276 + 0.9 \cdot 1/4 \cdot 7.6724 \\ &\qquad\qquad > 3 + 0.9 \cdot 1/4 \cdot 7.3276 + 0.9 \cdot 3/4 \cdot 7.6724\big\} \\ &= 1 + \mathbb{1}\big\{7.6724 > 9.8276\big\} = 1\end{aligned}$$

  - Notice that policy $\mu_2 = \mu_1$! So, we terminate.
- μ_1, μ_2 are optimal policies and $J^* = J_{\mu_1}$

Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

Telecom Paris, 18 December 2020

Lecture 4– LP Approach to Discounted Infinite-Horizon Dynamic Programming

Review of Lecture 3: The Discounted Case

- *Time-invariant* discrete-time dynamic system:

$$X_{k+1} = f(X_k, U_k, W_k), \qquad k = 0, 1, 2, \dots,$$

- Bounded time-invariant cost function $g(x, u, w) \in [-M, M]$

- Optimal discounted infinite-horizon cost:

$$J^*(a) := \min_{\pi} \mathbb{E}_{\{W_k\}} \left[\sum_{k=0}^{\infty} \gamma^k g(X_k, \mu_k(X_k), W_k) \middle| X_0 = a \right]$$

- Bellman's Equation: Optimal cost function $J^*(a)$ satisfies

$$J^*(a) = \min_{\mu} \mathbb{E}_W [g(a, \mu(a), W) + \gamma J^*(f(a, \mu(a), W))], \quad \forall a \in \mathcal{X}.$$

Review of Lecture 3, continued

- Value iteration algorithm based on the fact:

$$\lim_{k\to\infty} V_k(a) = J^*(a),$$

for any starting vector V_0 and

$$V_{k+1}(a) = \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V_k(f(a,\mu(a),W))], \quad k = 0,1,2,\ldots \tag{1}$$

$\rightarrow$ Start with $V_0 = \mathbf{0}$ and apply iteration (1) until satisfied with precision

- Policy iteration algorithm based on the following fact: If

$$\mathbb{E}_W[g(a,\mu_{k+1}(a),W)+\gamma J_{\mu_k}(a,\mu_{k+1}(a),W)] = \min_u \mathbb{E}_W\left[g(a,u)+\gamma J_{\mu_k}(f(a,u,W))\right], \tag{2}$$

then $J_{\mu_{k+1}}(a) \leq J_{\mu_k}(a), \quad \forall a \in \mathcal{X}.$

$\rightarrow$ Start with any policy μ_0, and apply policy iteration in (2)

Dynamic Programming Operator and Monotonicity

Definition (Dynamic Programming Operator)

Operator $\mathbb{T}$ (or $\mathbb{T}_{f,g,\gamma}$) acts on vector $V \in \mathcal{R}^{|\mathcal{X}|}$ componentwise as:

$$(\mathbb{T}V)(a) := \min_{\mu} \mathbb{E}_W[g(a,\mu(a),W) + \gamma V(f(a,\mu(a),W))], \quad \forall a \in \mathcal{X}.$$

- Monotonicity of $\mathbb{T}$: If $V(a) \leq (\mathbb{T}V(a)$ for all $a \in \mathcal{X}$, then

$$V(a) \leq (\mathbb{T}V)(a) \leq (\mathbb{T}^2V)(a) \leq \cdots J^*(a) \tag{3}$$

- The optimal cost vector J^* satisfies (3) by Bellman's equation: $(\mathbb{T}J^*) = J^*$

- Thus J^* is the largest vector satisfying $V(a) \leq (\mathbb{T}V)(a)$ for all $a \in \mathcal{X}$.

- Since $\mathbb{T}$ contains a min, $V(a) \leq (\mathbb{T}V)(a)$ is equivalent to:

$$V(a) \leq \mathbb{E}_W[g(a,\mu(a),W) + \gamma V(f(a,\mu(a),W))], \quad \forall a \in \mathcal{X}, \text{ and } \forall \mu.$$

Linear Programming Approach to find Vector J^*

- Let $\mathcal{X} = \{1, \ldots, m\}$ and $J(i) = J_i$.
- Pick positive weights $p_0(1), \ldots, p_0(m)$ summing to 1 and solve

Linear Programming Optimization Problem

$$\max_{J_1,\ldots,J_m} \quad (1-\gamma)\sum_{i=1}^{m} p_0(i) J_i$$

subject to:

$$J_i \leq \mathbb{E}_W\left[g(i,u,W)\right] + \gamma \cdot \sum_{j=1}^{m} P_{u,ij} J_j, \qquad \forall i, u$$

where $P_{u,ij} := \Pr[f(i,u,W) = j]$

(*Indices i and j were mixed up in the previous version of the slides!*
Also, we used policy μ instead of action u. We can use a single action u because for each i the constraint only depends on the single action in state i)

- Problem: the number of constraints can be huge.

Basic Optimization Theory: Primal-Dual LP Problems

Primal Problem

$$\max_{x_1,\ldots,x_n} \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{i,j} x_j \leq b_i, \quad i = 1, \ldots, m$$

Dual Problem

$$\min_{\lambda_1,\ldots,\lambda_m} \sum_{i=1}^{m} b_i \lambda_i$$

subject to

$$\sum_{i=1}^{m} a_{i,j} \lambda_i = c_j, \quad j = 1, \ldots, n$$

$$\lambda_i \geq 0, \quad i = 1, \ldots, m$$

- Solution has at most L non-degenerate components (i.e., components satisfying the constraints with strict inequalities)

The Dual Optimization Problem to the LP on the Previous Slide

Dual Problem

$$\min_{\{\rho(i,u)\}} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W\left[g(i,u,W)\right] \cdot \rho(i,u)$$

subject to:

$$\sum_u \rho(i,u) - \sum_{j=1}^{m} \sum_u \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma)p_0(i), \qquad \forall i = 1,\ldots,m \tag{4}$$

where $P_{u,ij} := \Pr[f(i,u,W) = j]$ and $\rho(i,u) \geq 0$ for all i, u.

- Solutions of linear programs are at the extreme points (corner points) of the intersection plane defined by the m constraints (4)
  $\rightarrow \exists$ an optimal solution $\rho^*(i,u)$ with only m components $\rho^*(i,u) > 0$
- If $\rho(i,u) = 0 \; \forall u$ for a specific i, then (4) cannot be satisfied for this i (the two sides (4) have different signs for constraint i)

$\Rightarrow$ For each $i = 1,\ldots,m$ there is exactly one $\rho^*(i,u) > 0$
There exists an optimal *stationary deterministic* policy $\mu^*(u|i) = \frac{\rho^*(i,u)}{\sum_v \rho^*(i,v)}$

The Dual Optimization Problem to the LP on the Previous Slide

Dual Problem

$$\min_{\{\rho(i,u)\}} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W\left[g(i,u,W)\right] \cdot \rho(i,u)$$

subject to:

$$\sum_{u} \rho(i,u) - \sum_{j=1}^{m} \sum_{u} \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma) p_0(i), \qquad \forall i = 1,\ldots,m \tag{4}$$

where $P_{u,ij} := \Pr[f(i,u,W) = j]$ and $\rho(i,u) \geq 0$ for all i, u.

- Summing both sides of (4) over $i = 1,\ldots,m$ shows that for any feasible $\rho(i,u)$:

$$\sum_{i=1}^{m} \sum_{u} \rho(i,u) = \sum_{i=1}^{m} p_0(i) = 1,$$

  So any feasible $\rho(i,u)$ can be a probability distribution over the states and actions.

Randomized Policies

- *A stationary randomized* policy μ chooses action $U_k = u$ with probability $\mu(u|i)$ when $X_k = i$

- We start with a random initial state $X_0 \sim p_0$ and calculate the *expected* discounted cost of this randomized policy

$$\begin{aligned} J_\mu(p_0) \quad &:= \quad \lim_{N\to\infty} \sum_{k=0}^{N} \gamma^k \mathbb{E}\Big[g(X_k, \mu(X_k), W) \Big] \\ &= \quad \lim_{N\to\infty} \sum_{k=0}^{N} \sum_{w} \sum_{i=1}^{m} \sum_{u} \gamma^k g(i,u,w) \mu(u|i) P_{X_k}(i) P_W(w), \end{aligned}$$

  whre $P_{X_k}(i)$ depends on the initial distribution p_0, and of course the stationary randomized policy μ and the state-transition function $f(\cdot,\cdot,\cdot)$.

State-Action Frequencies (also called Occupation Measures)

- Given an infinite-horizon policy π and initial state-distribution $p_0(i) = \Pr[X_0 = i]$, define the *state-action frequency*:

$$\rho^{\pi}_{p_0}(i,u) := (1-\gamma)\sum_{k=0}^{\infty} \gamma^k P^{\pi}_{p_0,k}(i,u), \quad i = 1,\ldots,m,$$

where $P^{\pi}_{p_0,k}(i,u) = \Pr[X_k = i, U_k = u]$ under policy π and initial state-distribution p_0.

- Define the *state-frequency*

$$\rho^{\pi}_{p_0}(i) := \sum_u \rho^{\pi}_{p_0}(i,u) = (1-\gamma)\sum_{k=0}^{\infty} \gamma^k P^{\pi}_{p_0,k}(i), \quad i = 1,\ldots,m,$$

- Under policy π and initial state-distribution p_0:

$$= (1-\gamma)J_{\pi}(p_0) \qquad = (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathbb{E}[g(X_k,U_k,W_k)]$$

$$= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \sum_{i,u} \mathbb{E}[g(i,u,W_k)] P^{\pi}_{p_0,k}(i,u)$$

$$= (1-\gamma)\sum_{i,u} \mathbb{E}[g(i,u,W_k)] \sum_{k=0}^{\infty}\gamma^k P^{\pi}_{p_0,k}(i,u) = \sum_{i,u} \mathbb{E}[g(i,u,W_k)]\rho^{\pi}_{p_0}(i,u).$$

Stationary Randomized Policy Deduced from State-Action Frequencies

- Given π, define a *stationary randomized* policy $\tilde{\pi} = (\mu^{\pi}_{p_0}, \mu^{\pi}_{p_0}, \ldots,)$ as

$$\mu^{\pi}_{p_0}(u|i) := \frac{\rho^{\pi}_{p_0}(i,u)}{\rho^{\pi}_{p_0}(i)}, \qquad \text{if } \rho^{\pi}_{p_0}(i) > 0,$$

  and $\mu^{\pi}_{p_0}(u|i)$ arbitrary if $\rho^{\pi}_{p_0}(i) = 0$. (*From any state-action frequencies $\rho(i,u) > 0$ one can derive a stationary policy.*)

- Under policy $\mu = \mu^{\pi}_{p_0}$ (proof on next slide):

$$\rho^{\mu}_{p_0}(i,u) = \rho^{\pi}_{p_0}(i,u), \qquad \forall i, u$$

- Therefore:

$$(1-\gamma)J_{\mu}(p_0) = \sum_{i,u} \mathbb{E}[g(i,u,W_k)]\rho^{\mu}_{p_0}(i,u)$$
$$= \sum_{i,u} \mathbb{E}[g(i,u,W_k)]\rho^{\pi}_{p_0}(i,u) = (1-\gamma)J_{\pi}(p_0)$$

$\Rightarrow$ For any π there is an equally-good *stationary randomized* policy μ
$\Rightarrow$ Without loss in performance one can restrict to stationary policies

Proof that $\rho^{\mu}_{p_0}(i,u) = \rho^{\pi}_{p_0}(i,u)$

$$
\begin{aligned}
&(1-\gamma)^{-1}\rho^{\pi}_{p_0}(i) \\
&= \sum_{k=0}^{\infty} \gamma^k P^{\pi}_{p_0,k}(i) = p_0(i) + \sum_{k=1}^{\infty} \gamma^k P^{\pi}_{p_0,k}(i) \\
&\overset{k'=k-1}{=} p_0(i) + \gamma \sum_{k'=0}^{\infty} \gamma^{k'} P^{\pi}_{p_0,k'+1}(i) \\
&= p_0(i) + \gamma \sum_{k'=0}^{\infty} \gamma^{k'} \Pr[X_{k'+1} = i] \\
&= p_0(i) + \gamma \sum_{k'=0}^{\infty} \gamma^{k'} \sum_{j,u} \Pr_{\pi}[X_{k'} = j, U_{k'} = u] \cdot \Pr[X_{k'+1} = i | X_{k'} = j, U_{k'} = u] \\
&= p_0(i) + \gamma \sum_{j,u} \sum_{k'=0}^{\infty} \gamma^{k'} \Pr_{\pi}[X_{k'} = j, U_{k'} = u] \cdot P_{u,ji} \\
&= p_0(i) + \frac{\gamma}{1-\gamma} \sum_{j,u} \rho^{\pi}_{p_0}(j,u) \cdot P_{u,ji} \qquad (5) \\
&= p_0(i) + \frac{\gamma}{1-\gamma} \sum_{j} \rho^{\pi}_{p_0}(j) \cdot \underbrace{\sum_{u} \mu(u|j) \cdot P_{u,ji}}_{=P_{\mu,ji}} = p_0(i) + \frac{\gamma}{1-\gamma} \sum_{j} \rho^{\pi}_{p_0}(j) \cdot P_{\mu,ji}
\end{aligned}
$$

Proof that $\rho^{\mu}_{p_0}(i,u) = \rho^{\pi}_{p_0}(i,u)$ continued

- Vectors $\boldsymbol{\rho}^{\pi}_{p_0} := (\rho^{\pi}_{p_0}(1), \dots, \rho^{\pi}_{p_0}(m))$ and $\mathbf{p}_0 := (p_0(1), \dots, p_0(m))$
(*Attention: changed to row-vectors for simplicity.*)

- P_μ the matrix with row-j and column-i entry equal to $P_{\mu,ji}$

- Then:

$$\boldsymbol{\rho}^{\pi}_{p_0} = (1-\gamma)\mathbf{p}_0 + \gamma \boldsymbol{\rho}^{\pi}_{p_0} \mathsf{P}_\mu$$

- Therefore:

$$\boldsymbol{\rho}^{\pi}_{p_0} = (1-\gamma)\mathbf{p}_0 \Big(\mathsf{I} - \gamma \mathsf{P}_\mu\Big)^{-1} = (1-\gamma)\mathbf{p}_0 \cdot \sum_{k=0}^{\infty} \gamma^k \mathsf{P}_\mu^k = (1-\gamma) \sum_{k=0}^{\infty} \gamma^k \mathbf{P}^{\mu}_{p_0,k} = \boldsymbol{\rho}^{\mu}_{p_0},$$

where $\mathbf{P}^{\mu}_{p_0,k}$ is the vector with i-th entry equal to $P^{\mu}_{p_0,k}(i)$.

Proof that $\rho^{\mu}_{p_0}(i,u) = \rho^{\pi}_{p_0}(i,u)$ continued II

- At the end of the previous slide we proved that the policies π and μ have same state-frequencies:

$$\rho^{\pi}_{p_0}(i) = \rho^{\mu}_{p_0}(i), \qquad \forall i.$$

- We now prove that the two policies also have same state-action frequencies:

$$\begin{aligned}\rho^{\pi}_{p_0}(i,u) &= \rho^{\pi}_{p_0}(i)\mu(u|i) = \rho^{\mu}_{p_0}(i)\mu(u|i) \\ &= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathrm{Pr}_{\mu}[X_k = i]\mu(u|i) \\ &= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathrm{Pr}_{\mu}[X_k = i, U_k = u] = \rho^{\mu}_{p_0}(i,u)\end{aligned}$$

State-Action Frequencies are the Variables in the Dual Problem, Slide 7

For any stationary policy μ, the state-action frequencies are feasible variables for the dual problem on slide 7 because $\rho^{\mu}_{p_0}(i, u) > 0$ and by eq. (5) on slide 11:

$$\underbrace{\sum_u \rho^{\mu}_{p_0}(i, u)}_{=\rho^{\mu}_{p_0}(i)} - \sum_{j=1}^{m} \sum_u \gamma \rho^{\mu}_{p_0}(j, u) P_{u,ji} = (1-\gamma) p_0(i), \qquad \forall i, \tag{6}$$

Moreover,

$$(1-\gamma) J_{\mu}(p_0) = \sum_{i,u} \mathbb{E}[g(i, u, W)] \rho^{\mu}_{p_0}(i, u)$$

and thus minimizing above right-hand side over all $\rho(i, u)$ satisfying (6) yields the minimum discounted infinite-horizon cost $J^*(p_0)$. (Recall that for any $\rho(i, u) > 0$ satisfying (6), it is possible to find a corresponding stationary policy μ s.t., $\rho(i, u)$ are the state-action frequencies of μ.)

Dual variables can be interpreted as the state-action frequencies!

Adding Constraints

- Can add a constraints on the cost to the linear programme on slide 6!
- Determininistic policies might not be optimal anymore, but randomized policies can have better performances.

Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

Telecom Paris, 18 December 2020

Lecture 5 – Multi-Armed Bandits and Unbounded Costs

Problems with Retirement Option

- Consider an infinite-horizon problem with bounded cost-per-stage $|g(a, u, w)| \leq M$, where at each stage k one can retire at cost $\gamma^k \cdot M_\infty$.

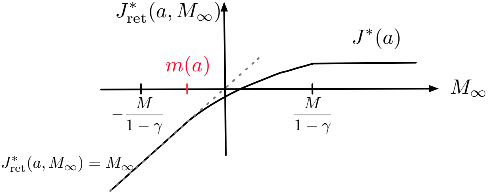
- Let $J^*_{\text{ret}}(a, M_\infty)$ be the optimal cost function for this problem. It satisfies the modified Bellman equation:

$$J^*_{\text{ret}}(a, M_\infty) = \min\left\{ M_\infty, \min_{\mu} \mathbb{E}_W \left[g(a, \mu(a), W) + \gamma J^*_{\text{ret}}\Big(f(a, \mu(a), W), M_\infty \Big) \right] \right\}.$$

- If $M_\infty \geq \frac{1}{1-\gamma} M$, then never retire

- If $M_\infty \leq -\frac{1}{1-\gamma} M$, then retire immediately

Optimal Policy under a Retirement Option



- Define

$$m(a) := \max\left\{M' : J^*_{\text{ret}}(a, M') = M'\right\}$$

Optimal Policy

Assume at stage k we have $X_k = a$.

- Retire if

$$m(a) \geq M_\infty,$$

- If $m(a) < M_\infty$, then play the optimal policy from Bellman's equation

Multi-Armed Bandits with Known Behaviours/Scheduling Projects

- Consider now L different DP problems $X_0^\ell, X_1^\ell, X_2^\ell, \ldots$ with different state evolution and cost functions $f^\ell(a, u, w)$ and $g^\ell(a, u, w)$, for $\ell = 1, \ldots, L$

- At each stage k one can retire at cost $\gamma^k \cdot M_\infty$

- Initial state $\mathbf{x}_0 = (x_0^1, x_0^2, \ldots, x_0^L)$

- At each stage k, retire or choose a project $\ell_k^* \in \{1, \ldots, L\}$ and an action u. If you don't retire:

$$X_{k+1}^{\ell_k^*} = f^{\ell_k^*}\left(X_k^{\ell^*}, u, W\right) \qquad \text{and} \qquad X_{k+1}^\ell = X_k^\ell, \;\; \forall \ell \in \{1, \ldots, L\} \backslash \{\ell_k^*\},$$

  and the stage-k cost is given by

$$g(x_1, \ldots, x_L, (u, \ell_k^*), W) = g^{\ell_k^*}(x_{\ell^*}, u, W).$$

- Wish to maximize the infinite-horizon discounted cost until retirement (if the player retires at all)

Optimal Scheduling Policy for Multi-Armed Bandit Problems

- Calculate the retirement threshold $m^{\ell}(a)$ for each project $\ell = 1, \ldots, L$ and state $a \in \mathcal{X}$ as explained before

Optimal Policy

Assume that at time k the states of the L projects are $x_1, \ldots, x_L$.

- Retire if

$$m^{\ell}(x_\ell) \geq M_\infty, \quad \forall \ell \in \{1, \ldots, L\}.$$

- Otherwise choose (ties can be split arbitrary)

$$\ell_k^* = \text{argmin}_\ell \;\; m^{\ell}(x_\ell)$$

and play the optimal policy for this project ℓ_k^* according to Bellman's equation.

Unbounded but Positive Costs

- Positive (possibly unbounded) costs $g(x, u, w) \in [0, \infty)$
- Discount factor $\gamma < 1$
- Bellman's equation remains valid:

$$J^* = TJ^*.$$

  But the solution might not be unique.
  The optimal cost function is given by the *smallest* fix-point!

- Value-iteration algorithm still works and provides optimal cost and optimal stationary policy!

  $\rightarrow$ finite-horizon solutions converge to the infinite-horizon solutions

- Policy iteration algorithm does not necessarily converge to optimal solution

The Quadratic Gaussian Case

- Vector states $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots \in \mathbb{R}^n$ and actions $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots \in \mathbb{R}^m$
- i.i.d. Gaussian noise vectors $\mathbf{W}_k$ of covariance matrix K_w
- State evolution when noise $\mathbf{W}_k = \mathbf{w}_k$ and controls $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots,$

  $$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = \mathsf{A}\mathbf{x}_k + \mathsf{B}\mathbf{u}_k + \mathbf{w}_k, \quad k = 0, 1, 2, \ldots$$

  for given matrices A and B.
- Deterministic cost function

  $$\sum_{k=0}^{\infty} \gamma^k g(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) = \sum_{k=0}^{\infty} \gamma^k \left(\mathbf{x}_k^{\mathsf{T}} \mathsf{Q} \mathbf{x}_k + \mathbf{u}_k^{\mathsf{T}} \mathsf{R} \mathbf{u}_k \right).$$
- Let R and Q be positive semi-definite.

Value-Iteration Algorithm on the Quadratic Gaussian Case

- Value-Iteration update rule for $k = 1, 2, \ldots$

$$V_k(\mathbf{x}) = \min_{\mu} \mathbb{E}_{\mathbf{W}}\Big[g(\mathbf{x}, \mu(\mathbf{x}), \mathbf{W}) + \gamma V_{k-1}\big(f(\mathbf{x}, \mu(\mathbf{x}), \mathbf{W})\big)\Big]$$
$$= \min_{\mathbf{u}} \left[\mathbf{x}^{\mathsf{T}}\mathrm{Q}\mathbf{x} + \mathbf{u}^{\mathsf{T}}\mathrm{R}\mathbf{u} + \gamma\mathbb{E}\Big[V_{k-1}\big(\mathrm{A}\mathbf{x} + \mathrm{B}\mathbf{u} + \mathbf{W}\big)\Big]\right]$$

- Start with $\mathbf{V}_0(\mathbf{x}) = 0$, for all vectors $\mathbf{x}$

- Notice that because R is positive semi-definite, $\mathbf{u}^{\mathsf{T}}\mathrm{R}\mathbf{u} \geq 0$ with equality for $\mathbf{u} = \mathbf{0}$. Thus:

$$\mathbf{V}_1(\mathbf{x}) = \min_{\mathbf{u}} \mathbf{x}^{\mathsf{T}}\mathrm{Q}\mathbf{x} + \mathbf{u}^{\mathsf{T}}\mathrm{R}\mathbf{u} = \mathbf{x}^{\mathsf{T}}\mathrm{Q}\mathbf{x}.$$

- For $k = 2$:

$$\mathbf{V}_2(\mathbf{x}) = \min_{\mathbf{u}} \left[\mathbf{x}^{\mathsf{T}}\mathrm{Q}\mathbf{x} + \mathbf{u}^{\mathsf{T}}\mathrm{R}\mathbf{u} + \gamma\mathbb{E}_W\left[(\mathbf{x}^{\mathsf{T}}\mathrm{A}^{\mathsf{T}} + \mathbf{u}\mathrm{B}^{\mathsf{T}} + \mathbf{W}^{T})\mathrm{Q}(\mathbf{W} + \mathrm{B}\mathbf{u} + \mathrm{A}\mathbf{x})\right]\right]$$
$$= \mathbf{x}^{\mathsf{T}}\mathrm{Q}\mathbf{x} + \gamma\mathbb{E}\left[\mathbf{W}^{T}\mathrm{Q}\mathbf{W}\right] + \min_{\mathbf{u}}\left[\mathbf{u}^{\mathsf{T}}\mathrm{R}\mathbf{u} + \gamma(\mathbf{x}^{\mathsf{T}}\mathrm{A}^{\mathsf{T}} + \mathbf{u}\mathrm{B}^{\mathsf{T}})\mathrm{Q}(\mathrm{B}\mathbf{u} + \mathrm{A}\mathbf{x})\right]$$
$$= \mathbf{x}^{\mathsf{T}} \underbrace{(\mathrm{Q} + \mathrm{A}^{\mathsf{T}}\mathrm{Q}\mathrm{A})}_{\text{positive semidefinite}} \mathbf{x} + \gamma\mathbb{E}\left[\mathbf{W}^{T}\mathrm{Q}\mathbf{W}\right]$$
$$+ \min_{\mathbf{u}}\left[\mathbf{u}^{\mathsf{T}} \underbrace{(\mathrm{R} + \gamma\mathrm{B}^{\mathsf{T}}\mathrm{Q}\mathrm{B})}_{\text{positive semidefinite}} \mathbf{u} + 2\gamma\mathbf{x}^{\mathsf{T}}\mathrm{A}^{\mathsf{T}}\mathrm{Q}\mathrm{B}\mathbf{u}\right]$$

Minimizing Quadratic Forms

- Consider the quadratic form in $\mathbf{u}$:

$$f(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathsf{T}}\mathsf{M}\mathbf{u} + \mathbf{c}^{\mathsf{T}}\mathbf{u},$$

  where $\mathbf{c}$ is an arbitrary vector and $\mathbf{M}$ is a positive semidefinite matrix. (This latter assumption is need to ensure convexity of the function f.)

- The gradient of f with respect to $\mathbf{u}$ is:

$$\nabla f(\mathbf{u}) = \mathsf{M}\mathbf{x} + \mathbf{c}.$$

- The function f is minimized for

$$\mathbf{u}^* = -\mathsf{M}^{-1}\mathbf{c}$$

  and the minimum value of f is

$$f_{\min} := \min_{\mathbf{u}} f(\mathbf{u}) = -\frac{1}{2}\mathbf{c}^{\mathsf{T}}\mathsf{M}^{-1}\mathbf{c}.$$

Quadratic Gaussian Example continued

- We obtain for $k=2$:

$$\begin{aligned}\mathbf{V}_2(\mathbf{x}) &= \mathbf{x}^{\mathsf{T}}(\mathsf{Q}+\gamma\mathsf{A}^{\mathsf{T}}\mathsf{Q}\mathsf{A})\mathbf{x}+\gamma\mathbb{E}\left[\mathbf{W}^{T}\mathsf{Q}\mathbf{W}\right]-\gamma^2\mathbf{x}^{\mathsf{T}}\mathsf{A}^{\mathsf{T}}\mathsf{Q}\mathsf{B}(\mathsf{R}+\gamma\mathsf{B}^{\mathsf{T}}\mathsf{Q}\mathsf{B})^{-1}\mathsf{B}^{\mathsf{T}}\mathsf{Q}\mathsf{A}\mathbf{x}\\ &= \gamma\mathbb{E}\left[\mathbf{W}^{T}\mathsf{Q}\mathbf{W}\right]+\mathbf{x}^{\mathsf{T}}\underbrace{\left(\mathsf{Q}+\gamma\mathsf{A}^{\mathsf{T}}\mathsf{Q}\mathsf{A}-\gamma^2\mathsf{A}^{\mathsf{T}}\mathsf{Q}\mathsf{B}(\mathsf{R}+\gamma\mathsf{B}^{\mathsf{T}}\mathsf{Q}\mathsf{B})^{-1}\mathsf{B}^{\mathsf{T}}\mathsf{Q}\mathsf{A}\right)}_{=:\mathsf{M}_2}\mathbf{x}\end{aligned}$$

- The optimal control is linear:

$$\mathbf{u}^{*}=-\gamma(\mathsf{R}+\gamma\mathsf{B}^{\mathsf{T}}\mathsf{Q}\mathsf{B})^{-1}\mathsf{B}^{\mathsf{T}}\mathsf{Q}\mathsf{A}\mathbf{x}$$

- $\mathbf{V}_2$ has a similar form to $\mathbf{V}_1$ but with M_2 (which is positive semi-definite, see slide 12) instead of Q, and there is an additional summand $\gamma\mathrm{tr}(\mathsf{K}_W\mathsf{Q})$
- Can obtain $\mathbf{V}_3$ following the same reasoning, but exchanging Q with M_2 and adding $\gamma\cdot\gamma\mathbb{E}\left[\mathbf{W}^{T}\mathsf{Q}\mathbf{W}\right]$ to the cost

Semi-positivity of matrix M_2

- By standard manipulations on matrices:

$$
\begin{aligned}
\Gamma &:= \gamma A^T QA - \gamma^2 A^T QB(R+\gamma B^T QB)^{-1} B^T QA \\
&= \gamma A^T \left(Q - \gamma QB(R+\gamma B^T QB)^{-1} B^T Q \right) A \\
&= \gamma A^T \left(QB(B^T QB)^{-1} B^T Q - \gamma QB(R+\gamma B^T QB)^{-1} B^T Q \right) A \\
&= \gamma A^T QB \left((B^T QB)^{-1} - \gamma (R+\gamma B^T QB)^{-1} \right) B^T QA \\
&= \gamma A^T QB \Big((B^T QB)^{-1} (R+\gamma B^T QB)(R+\gamma B^T QB)^{-1} \\
&\qquad\qquad - (B^T QB)^{-1} (B^T QB) \gamma (R+\gamma B^T QB)^{-1} \Big) B^T QA \\
&= \gamma A^T QB(B^T QB)^{-1} \Big((R+\gamma B^T QB) - \gamma (B^T QB) \Big) (R+\gamma B^T QB)^{-1} B^T QA \\
&= \gamma A^T QB(B^T QB)^{-1} R(R+\gamma B^T QB)^{-1} B^T QA
\end{aligned}
$$

- $\Gamma \succeq 0$ is positive semidefinite because: - Q, R are positive semidefinite and for any positive semidefinite matrices M, N and arbitrary matrix S: $M + N \succeq 0,\ M \cdot N \succeq 0,\ M^{-1} \succeq 0,\ S^T MS \succeq 0$ are also positive semidefinite.

- By the same reasons, also $M_2 = \Gamma + Q$ is positive semidefinite

Quadratic Gaussian Example continued II

- We obtain for $k = 3$:

$$\begin{aligned}
\mathbf{V}_3(\mathbf{x}) &= \min_{\mathbf{u}} \left[\mathbf{x}^{\mathsf{T}} \mathrm{Q} \mathbf{x} + \mathbf{u}^{\mathsf{T}} \mathrm{R} \mathbf{u} + \gamma \mathbb{E}_W \left[(\mathbf{x}^{\mathsf{T}} \mathrm{A}^{\mathsf{T}} + \mathbf{u} \mathrm{B}^{\mathsf{T}} + \mathbf{W}^T) \mathrm{M}_2 (\mathbf{W} + \mathrm{B}\mathbf{u} + \mathrm{A}\mathbf{x}) \right] \right] \\
&\quad + \gamma^2 \mathbb{E}\left[\mathbf{W}^T \mathrm{Q} \mathbf{W}\right] \\
&= \mathbf{x}^{\mathsf{T}} (\mathrm{Q} + \gamma \mathrm{A}^{\mathsf{T}} \mathrm{M}_2 \mathrm{A}) \mathbf{x} + \gamma^2 \mathbb{E}\left[\mathbf{W}^T \mathrm{Q} \mathbf{W}\right] + \gamma \mathbb{E}\left[\mathbf{W}^T \mathrm{M}_2 \mathbf{W}\right] \\
&\quad + \min_{\mathbf{u}} \left[\mathbf{u}^{\mathsf{T}} (\mathrm{R} + \gamma \mathrm{B}^{\mathsf{T}} \mathrm{M}_2 \mathrm{B}) \mathbf{u} + 2\gamma \mathbf{x}^{\mathsf{T}} \mathrm{A}^{\mathsf{T}} \mathrm{M}_2 \mathrm{B} \mathbf{u} \right] \\
&= \gamma^2 \mathbb{E}\left[\mathbf{W}^T \mathrm{Q} \mathbf{W}\right] + \gamma \mathbb{E}\left[\mathbf{W}^T \mathrm{M}_2 \mathbf{W}\right] \\
&\quad + \mathbf{x}^{\mathsf{T}} \underbrace{\left(\mathrm{Q} + \gamma \mathrm{A}^{\mathsf{T}} \mathrm{M}_2 \mathrm{A} - \gamma^2 \mathrm{A}^{\mathsf{T}} \mathrm{M}_2 \mathrm{B} (\mathrm{R} + \gamma \mathrm{B}^{\mathsf{T}} \mathrm{M}_2 \mathrm{B})^{-1} \mathrm{B}^{\mathsf{T}} \mathrm{M}_2 \mathrm{A} \right)}_{=:\mathrm{M}_3} \mathbf{x}
\end{aligned}$$

- The optimal control is linear:

$$\mathbf{u}^* = -\gamma (\mathrm{R} + \gamma \mathrm{B}^{\mathsf{T}} \mathrm{M}_2 \mathrm{B})^{-1} \mathrm{B}^{\mathsf{T}} \mathrm{M}_2 \mathrm{A} \mathbf{x}$$

- Can obtain $\mathbf{V}_4$ following the same reasoning, but exchanging M_2 with M_3 and adding $\gamma \cdot \left(\gamma^2 \mathbb{E}\left[\mathbf{W}^T \mathrm{Q} \mathbf{W}\right] + \gamma \mathbb{E}\left[\mathbf{W}^T \mathrm{M}_2 \mathbf{W}\right]\right)$ to the cost. ETC.

Quadratic Gaussian Example continued III

- Continuing along the same lines, we observe:

$$\mathbf{V}_k(\mathbf{x}) = \sum_{\ell=1}^{k-1} \gamma^{k-\ell} \mathbb{E}\left[\mathbf{W}^T \mathrm{M}_\ell \mathbf{W}\right] + \mathbf{x}^\mathsf{T} \mathrm{M}_k \mathbf{x},$$

  where $\mathrm{M}_1 = \mathrm{Q}$ and for $k = 2, 3, \ldots$:

$$\begin{aligned} \mathrm{M}_k &= \mathrm{Q} + \gamma \mathrm{A}^\mathsf{T} \mathrm{M}_{k-1} \mathrm{A} - \gamma^2 \mathrm{A}^\mathsf{T} \mathrm{M}_{k-1} \mathrm{B} (\mathrm{R} + \gamma \mathrm{B}^\mathsf{T} \mathrm{M}_{k-1} \mathrm{B})^{-1} \mathrm{B}^\mathsf{T} \mathrm{M}_{k-1} \mathrm{A} \\ &= \mathrm{Q} + \tilde{\mathrm{A}}^\mathsf{T} \mathrm{M}_{k-1} \tilde{\mathrm{A}} - \tilde{\mathrm{A}}^\mathsf{T} \mathrm{M}_{k-1} \tilde{\mathrm{B}} (\mathrm{R} + \tilde{\mathrm{B}}^\mathsf{T} \mathrm{M}_{k-1} \tilde{\mathrm{B}})^{-1} \tilde{\mathrm{B}}^\mathsf{T} \mathrm{M}_{k-1} \tilde{\mathrm{A}}, \end{aligned}$$

  where $\tilde{\mathrm{A}} := \sqrt{\gamma}\mathrm{A}$ and $\tilde{\mathrm{B}} := \sqrt{\gamma}\mathrm{B}$

- It can again be shown that $\mathrm{M}_k \succeq 0$ is positive semidefinite.

- The sequence M_k is known to converge to M^* the solution of the *Algebraic Riccati Equation* (important in control theory)

$$\mathrm{M} = \mathrm{Q} + \tilde{\mathrm{A}}^\mathsf{T} \mathrm{M} \tilde{\mathrm{A}} - \tilde{\mathrm{A}}^\mathsf{T} \mathrm{M} \tilde{\mathrm{B}} (\mathrm{R} + \tilde{\mathrm{B}}^\mathsf{T} \mathrm{M} \tilde{\mathrm{B}})^{-1} \tilde{\mathrm{B}}^\mathsf{T} \mathrm{M} \tilde{\mathrm{A}}$$

  whenever the pair $(\tilde{\mathrm{A}}, \tilde{\mathrm{B}})$ is controllable and $(\tilde{\mathrm{A}}, \tilde{\mathrm{C}})$ is observable, where $\mathrm{Q} = \mathrm{C}^\mathsf{T}\mathrm{C}$.

Controllability and Observability

Definition (Controllability)

A pair (A, B), where A is an $n \times n$ matrix and B a $n \times m$ matrix, is said *controllable* if the $n \times nm$ matrix

$$[\mathsf{B}, \mathsf{AB}, \mathsf{A}^2\mathsf{B}, \ldots \mathsf{A}^{n-1}\mathsf{B}]$$

has full rank

Definition (Observability)

A pair (A, C) is said *observable* if the pair $(\mathsf{A}^\mathsf{T}, \mathsf{C}^\mathsf{T})$ is *controllable*.

The Solution of the Quadratic Gaussian Example

- Since M_ℓ converges, also the weighted sum of the noise-terms converges. Using the geometric sum formula:

$$\lim_{k\to\infty}\sum_{\ell=1}^{k-1}\gamma^{k-\ell}\mathbb{E}\left[\mathbf{W}^T\mathsf{M}_\ell\mathbf{W}\right]=\frac{1}{1-\gamma}\mathbb{E}\left[\mathbf{W}^T\mathsf{M}^*\mathbf{W}\right]$$

where M^* is the solution to the Algebraic Riccati equation

$$\mathsf{M}=\mathsf{Q}+\tilde{\mathsf{A}}^\mathsf{T}\mathsf{M}\tilde{\mathsf{A}}-\tilde{\mathsf{A}}^\mathsf{T}\mathsf{M}\tilde{\mathsf{B}}(\mathsf{R}+\tilde{\mathsf{B}}^\mathsf{T}\mathsf{M}\tilde{\mathsf{B}})^{-1}\tilde{\mathsf{B}}^\mathsf{T}\mathsf{M}\tilde{\mathsf{A}} \tag{1}$$

Optimal Infinite cost $J^*(\mathbf{x})$

For any state vector $\mathbf{x}$:

$$J^*(\mathbf{x})=\frac{1}{1-\gamma}\mathbb{E}\left[\mathbf{W}^T\mathsf{M}^*\mathbf{W}\right]+\mathbf{x}^\mathsf{T}\mathsf{M}^*\mathbf{x}.$$

where M^* is the solution to (1) english

Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

Telecom Paris, 8 Jan 2021

Lecture 6– Constrained Discounted Problems and Average-Cost Problems

Outlook Today

- *Time-invariant* discrete-time dynamic system:

$$X_{k+1} = f(X_k, U_k, W_k), \qquad k = 0, 1, 2, \ldots,$$

  disturbance $\{W_k\}$ i.i.d.

- Bounded time-invariant cost function $g(x, u, w) \in [-M, M]$

- Optimal discounted infinite-horizon cost:

$$\bar{J}^*(p_0) := \min_{\pi} \lim_{N\to\infty} \mathbb{E}_{X_0,\{W_k\}}\left[\sum_{k=0}^{N-1} \gamma^k g(X_k, \mu_k(X_k), W_k)\right]$$

- Today we add cost constraints: A policy π is admissible only if

$$\mathbb{E}^{\pi}_{X_0,\{W_k\}}\left[\sum_{k=0}^{\infty} \gamma^k d_\ell(X_k, \mu_k(X_k), W_k)\right] \le D_\ell, \qquad \ell = 1, \ldots, L.$$

Outlook Today

- *Time-invariant* discrete-time dynamic system:

$$X_{k+1} = f(X_k, U_k, W_k), \qquad k = 0, 1, 2, \ldots,$$

  disturbance $\{W_k\}$ i.i.d.

- Bounded time-invariant cost function $g(x, u, w) \in [-M, M]$

- Optimal average infinite-horizon cost:

$$\bar{J}^*(p_0) := \min_{\pi} \lim_{N \to \infty} \mathbb{E}_{X_0, \{W_k\}} \left[\sum_{k=0}^{N-1} \frac{1}{N} g(X_k, \mu_k(X_k), W_k) \right]$$

Review of Lecture 4: LP Programming Approach

Primal Problem

$$\max_{J_1,\dots,J_m} \quad (1-\gamma)\sum_{i=1}^{m} p_0(i) J_i$$

subject to:

$$J_i \leq \mathbb{E}_W\big[g(i,u,W)\big] + \gamma \cdot \sum_{j=1}^{m} P_{u,ij} J_j, \qquad \forall i, u$$

where $P_{u,ij} := \Pr[f(i,u,W) = j]$

Review of Lecture 4: LP Programming Approach

Dual Problem

$$\min_{\{\rho(i,u)\}} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W\left[g(i,u,W)\right] \cdot \rho(i,u)$$

subject to:

$$\sum_{u} \rho(i,u) - \sum_{j=1}^{m} \sum_{u} \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma)p_0(i), \qquad i = 1, \ldots, m$$

where $P_{u,ij} := \Pr[f(i,u,W) = j]$ and $\rho(i,u) \geq 0$ for all i, u.

- State-action frequencies/occupation measures $\rho(i,u)$ form a pmf and determine a randomized stationary policy $\mu(u|i) = \frac{\rho(i,u)}{\sum_u \rho(i,u)}$
- $\exists$ an optimal $\rho^*(i,u) > 0$ with only m components, one for each state i $\Longrightarrow$ Deterministic stationary policies are optimal!

Constrained Discounted Infinite-Horizon Problems

- *Time-invariant* discrete-time dynamic system:

$$X_{k+1} = f(X_k, U_k, W_k), \qquad k = 0, 1, 2, \ldots,$$

- Bounded time-invariant cost function $g(x, u, w) \in [-M, M]$ and constraint-cost functions $d_\ell(x, u, w)$, for $\ell = 1, \ldots, L$, as well as maximum constraints $D_1, \ldots, D_L$

- Optimal discounted infinite-horizon cost:

$$J^*(a) := \min_{\pi} \lim_{N \to \infty} \mathbb{E}_{X_0, \{W_k\}} \left[\sum_{k=0}^{N} \gamma^k g(X_k, \mu_k(X_k), W_k) \right]$$

where minimum is over all policies $\pi = (\mu_1, \mu_2, \ldots)$ satisfying

$$\lim_{N \to \infty} \mathbb{E}_{X_0, \{W_k\}} \left[\sum_{k=0}^{N} \gamma^k d_\ell(X_k, \mu_k(X_k), W_k) \right] \le D_\ell, \qquad \ell = 1, \ldots, L.$$

Can express constraints using State-Action Frequencies

For all $\ell = 1, \ldots, L$:

$$
\begin{aligned}
&(1-\gamma)\mathbb{E}_{X_0,\{W_k\}}\left[\sum_{k=0}^{\infty}\gamma^k d_\ell(X_k, \mu_k(X_k), W_k)\right] \\
&= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \sum_{i,u}\mathbb{E}\big[d_\ell(i,u,W_k)\big]\Pr[X_k = i, \mu_k(i) = u] \\
&= \sum_{i,u}\mathbb{E}\big[d_\ell(i,u,W)\big](1-\gamma)\sum_{k=0}^{\infty}\gamma^k \Pr[X_k = i, \mu_k(i) = u] \\
&= \sum_{i,u}\mathbb{E}[d_\ell(i,u,W)]\rho(i,u) \\
&\leq (1-\gamma)D_\ell.
\end{aligned}
$$

Dual Linear Programming Problem with Constraints

Dual Linear Programming Problem for Constrained Optimization Problem

$$J^*(p_0) = \min_{\rho(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W\left[g(i,u,W)\right] \cdot \rho(i,u)$$

subject to:

$$\sum_u \rho(i,u) - \sum_{j=1}^{m} \sum_u \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma) p_0(i), \qquad i = 1, \ldots, m,$$

and

$$\sum_{i,u} \mathbb{E}[d_\ell(i,u,W)]\rho(i,u) \leq (1-\gamma) D_\ell, \quad \ell = 1, \ldots, L.$$

- Optimal policy is generally stationary with $\leq L$ randomized actions

Dual Problem with Constraints → Lagrange Multipliers

Dual Problem for Constrained Optimization Problem

$$J^*(p_0) = \min_{\rho(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W\left[\underbrace{g(i,u,W)}\right] \cdot \rho(i,u)$$

subject to:

$$\sum_{u} \rho(i,u) - \sum_{j=1}^{m} \sum_{u} \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma) p_0(i), \qquad i = 1, \ldots, m,$$

and

$$\sum_{i,u} \mathbb{E}[d_\ell(i,u,W)] \rho(i,u) \leq (1-\gamma) D_\ell, \quad \ell = 1, \ldots, L.$$

- Add additional constraints using Lagrange Multipliers $\lambda_1, \ldots, \lambda_L$!

Dual Problem with Constraints → Lagrange Multipliers

Dual Problem for Constrained Optimization Problem

$$J^*(p_0) = \min_{\rho(i,u)\geq 0} \sup_{\lambda_1,\ldots,\lambda_L\geq 0} \sum_{i=1}^{m}\sum_{u} \mathbb{E}_W \underbrace{\Big[g(i,u,W) + \sum_{\ell=1}^{L} \lambda_\ell d_\ell(i,u,W) \Big]}_{} \cdot \rho(i,u) - \sum_{\ell=1}^{L} \lambda_\ell D_\ell$$

subject to:

$$\sum_u \rho(i,u) - \sum_{j=1}^{m}\sum_u \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma) p_0(i), \qquad i = 1,\ldots,m,$$

- Add additional constraints using Lagrange Multipliers $\lambda_1,\ldots,\lambda_L$!

Dual Problem with Constraints → Lagrange Multipliers

Dual Problem for Constrained Optimization Problem

$$J^*(p_0) = \sup_{\lambda_1,\ldots,\lambda_L \geq 0} \min_{\rho(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \underbrace{\mathbb{E}_W\left[g(i,u,W) + \sum_{\ell=1}^{L} \lambda_\ell d_\ell(i,u,W) \right]}_{} \cdot \rho(i,u) - \sum_{\ell=1}^{L} \lambda_\ell D_\ell$$

subject to:

$$\sum_u \rho(i,u) - \sum_{j=1}^{m} \sum_u \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma)p_0(i), \qquad i = 1,\ldots,m,$$

- Add additional constraints using Lagrange Multipliers $\lambda_1, \ldots, \lambda_L$!
- Strong duality holds by standard arguments

Dual Problem with Constraints → Lagrange Multipliers

Dual Problem for Constrained Optimization Problem

$$J^*(p_0) = \sup_{\lambda_1,\ldots,\lambda_L \geq 0} \min_{\rho(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W \Big[\underbrace{g(i,u,W) + \sum_{\ell=1}^{L} \lambda_\ell d_\ell(i,u,W)}_{\text{new cost function } \tilde{g}(i,u,W)} \Big] \cdot \rho(i,u) - \sum_{\ell=1}^{L} \lambda_\ell D_\ell$$

subject to:

$$\sum_u \rho(i,u) - \sum_{j=1}^{m} \sum_u \gamma P_{u,ij} \cdot \rho(j,u) = (1-\gamma) p_0(i), \qquad i = 1, \ldots, m,$$

- Add additional constraints using Lagrange Multipliers $\lambda_1, \ldots, \lambda_L$!
- Strong duality holds by standard arguments
- For each $\lambda_1, \ldots, \lambda_L$: solve for the new cost function $\tilde{g}$
  → minimum achieved by a deterministic stationary policy (proof as before)

Optimal Average Cost Problems

- Optimal average infinite horizon cost:

$$\bar{J}^*(p_0) := \min_{\pi} \bar{J}^{\pi}(p_0)$$

  where for a given policy π:

$$\bar{J}^{\pi}(p_0) := \overline{\lim_{N\to\infty}} \frac{1}{N} \mathbb{E}_{X_0,\{W_k\}}\left[\sum_{k=0}^{N-1} g(X_k, U_k, W_k)\right]$$

- We can again restrict to Markov policies because objective function only depends on $\{P_{X_k,U_k}\}_{k\geq 0}$ as in the discounted case

Unichain Assumption

- For a stationary policy μ, the induced Markov chain has transition matrix

$$\mathrm{P}_\mu(i,j) := \Pr[X_{k+1} = j | X_k = i] = \sum_u \mu(u|i) \Pr[f(i,u,W) = j].$$

- Recall: If a Markov chain is irreducible (i.e., $\mathcal{X}$ is a recurrent class) and aperiodic, its state-distribution tends to the unique stationary distribution, irrespective of the X_0-distribution.

- If the Markov chain is periodic, the distribution can "toggle" between different distributions

- The same holds also when there is an additional set of transient states. (At some point the Markov chain will end in the recurrent class and converge (or toggle).)

Definition (Unichain)

A Dynamic Programming Problem is called *Unichain* if the state space can be decomposed into $\mathcal{S} \cup \mathcal{T} = \mathcal{X}$, with $\mathcal{S} \cap \mathcal{T} = \emptyset$, so that for all stationary policies μ,the set $\mathcal{S}$ forms a recurrent class and $\mathcal{T}$ is a set of transient states.

Expressing the Cost-Function in State-Action Frequencies

- For a given policy π:

$$\bar{J}^{\pi}(p_0) := \overline{\lim_{N\to\infty}} \frac{1}{N} \mathbb{E}_{X_0,\{W_k\}}\left[\sum_{k=0}^{N-1} g(X_k, \mu_k(X_k), W_k)\right]$$
$$= \overline{\lim_{N\to\infty}} \sum_{i,u} \mathbb{E}[g(i,u,W)] \cdot \frac{1}{N}\sum_{k=0}^{N-1} \Pr[X_k = i, \mu_k(i) = u]$$
$$= \overline{\lim_{N\to\infty}} \sum_{i,u} \mathbb{E}[g(i,u,W)] \cdot \nu_N^{\pi}(i,u)$$

- *N-horizon state-action frequency*

$$\nu_N^{\pi}(i,u) := \frac{1}{N}\sum_{k=0}^{N-1} \Pr[X_k = i, \mu_k(i) = u]$$

- N-horizon state-action frequency (occupation measure) $\nu_N^{\pi}(i,u)$ describes the probability of observing the state-action pair (i,u) at a random time T which is uniform over $\{0, 1, \dots, N-1\}$

Convergence of $\nu_N^\pi(i,u)$

- Depending on the policy π, the sequences $\{\nu_N^\pi(i,u)\}_{N\geq 1}$ might diverge to various accumulation points! $\rightarrow$ therefore use limsup!

- Let ν^π be an accumulation point of $\{\nu_N^\pi(i,u)\}_{N\geq 1}$. Then (see next slide):

$$\sum_u \nu^\pi(i,u) = \sum_{j,u} \nu^\pi(j,u) P_{u,ji}$$

- Under the unichain assumption and stationary policy μ, the sequences $\{\nu_N^\mu(i,u)\}_{N\geq 1}$ converge to the (infinite-horizon) *state-action frequencies*

$$\nu_\infty^\mu(i,u) := \lim_{N\to\infty} \nu_N^\mu(i,u) = \xi^\mu(i)\cdot\mu(u|i),$$

  irrespective of p_0, and where $\xi^\mu = (\xi^\mu(1),\ldots,\xi^\mu(m))$ is the stationary distribution of the Markov chain P_μ.
  Proof: Apply Césaro's mean theorem and the limit $\Pr[X_k = i] \rightarrow \xi^\mu(i)$

Proof that $\sum_u v^\pi(i,u) = \sum_{j,u} v^\pi(j,u) P_{u,ji}$

Consider any initial distribution $p(0)$ and increasing sequence $\{N_l\}_{l\geq 0}$ such that $\nu^\pi_{N_l}(i,u)$ converges to $v^\pi(i,u)$ as $l \to \infty$ for all u, i.
For any $l > 0$:

$$\begin{aligned}
&\sum_v \nu^\pi_{N_l}(i,v) - \frac{1}{N_l} p(0) \\
&= \sum_v \frac{1}{N_l} \sum_{k=1}^{N_l-1} \Pr[X_k = i, \mu_k(i) = v] = \frac{1}{N_l} \sum_{k=1}^{N_l-1} \Pr[X_k = i] \\
&= \frac{1}{N_l} \sum_{k=1}^{N_l-1} \sum_{j,u} \Pr[X_{k-1} = j, U_{k-1} = u] P_{u,ji} \\
&= \frac{1}{N_l} \sum_{k'=0}^{N_l-2} \sum_{j,u} \Pr[X_{k'} = j, U_{k'} = u] P_{u,ji} \\
&= \frac{1}{N_l} \sum_{k'=0}^{N_l-1} \sum_{j,u} \Pr[X_{k'} = j, U_{k'} = u] P_{u,ji} - \frac{1}{N_l} \Pr[X_{N_l-1} = j, U_{N_l-1} = u] P_{u,ji}
\end{aligned}$$

Taking limits $l \to \infty$ and thus $N_l \to \infty$ on both sides, yields the desired expressions because the sums and the limit can be exchanged

Can restrict to Stationary Policies

- Given any policy π and accumulation point $\nu^{\pi}(i,u)$.
- Choose a stationary policy μ with

$$\mu(u|i) = \frac{\nu^{\mu}(i,u)}{\sum_v \nu^{\mu}(i,v)}.$$

- π and μ have same state-action frequencies:

$$\nu^{\pi}(i,u) = \mu(u|i)\cdot\Big(\sum_v \nu^{\mu}(i,v)\Big) = \underbrace{\mu(u|i)\xi^{\mu}(i)}_{=\nu^{\mu}_{\infty}(i,u)}\cdot\underbrace{\frac{\sum_v \nu^{\mu}(i,v)}{\xi^{\mu}(i)}}_{=1,\text{ see next slide}} = \nu^{\mu}_{\infty}(i,u)$$

- $\Rightarrow$ Cost function of μ at least as good as for π:

$$\bar{J}^{\pi} \geq \sum_{i,u}\mathbb{E}[g(i,u,W)]\cdot\nu^{\pi}(i,u) = \sum_{i,u}\mathbb{E}[g(i,u,W)]\cdot\nu^{\mu}_{\infty}(i,u) = \bar{J}^{\mu}$$

Can restrict to (randomized) stationary policies μ

Proof that $\sum_v \nu^\mu(i,v) = \xi^\mu(i)$

- We have

$$\nu^\pi(i) := \sum_u \nu^\pi(i,u) = \sum_{j,u} \nu^\pi(j,u) P_{u,ji} = \sum_j \nu^\pi(i) \sum_u \mu(u|j) P_{u,ji}$$
$$= \sum_j \nu^\pi(j) \mathsf{P}_{\mu,ji},$$

- Therefore $\boldsymbol{\nu}^\pi$ equals the unique stationary distribution $\boldsymbol{\xi}^\mu$ of the MC P_μ induced by action policy μ.

Linear Programme Solution based on State-Action Frequencies

- Since we can restrict to stationary distributions:

"Dual Problem" for Average Costs

$$\bar{J}^* = \min_{\nu(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W\left[g(i,u,W)\right] \cdot \nu(i,u)$$

subject to:

$$\sum_{v} \nu(i,v) = \sum_{j=1}^{m} \sum_{u} \nu(j,u) P_{u,ji} \qquad i = 1,\dots,m, \tag{1}$$

$$\sum_{i,u} \nu(i,u) = 1.$$

- m constraints are linearly dependent because both sides of (1) sum to 1.
  $\rightarrow$ Optimal $\nu^*(i,u) > 0$ for at most m pairs (i,u) (m lin. indep. constr.)

Deterministic stationary policy $\mu^*(u|i) = \frac{\nu^*(i,u)}{\sum_v \nu^*(i,v)}$ is optimal

Value-Iteration Algorithm to Find Optimal Average Cost

- Modified update operator $\mathbb{T}_{\text{avg}} \colon \mathbf{V} \mapsto \min_\mu \left[\mathbb{E}_W[g(i,\mu(i),W)] + \mathsf{P}_\mu \mathbf{V}\right]$

- A modified Bellman's equation holds

- For any initial vector $\mathbf{V}$:

$$\frac{1}{N}\mathbb{T}_{\text{avg}}^N \mathbf{V} \to \bar{J}^* \quad \text{as } N \to \infty.$$

- **Value-iteration algorithm:** Pick an arbitrary initial vector $\mathbf{J}_0$ and iterate until convergence:

$$\mathbf{J}_{k+1} = \frac{k}{k+1}\mathbb{T}_{\text{avg}}\mathbf{J}_k, \qquad k = 0, 1, \ldots,$$

Policy- Iteration Algorithm to Find Optimal Average Cost

- Modified operators $\mathbb{T}_{\text{avg}}$ and $\mathbb{T}_{\text{avg},\mu}$: $\mathbf{V} \mapsto \left[\mathbb{E}_W[g(i,\mu(i),W)] + \mathsf{P}_\mu \mathbf{V}\right]$

- **Policy-iteration algorithm:** use above operators and slightly modified policy evaluation step.

- Start with arbitrary initial policy μ_0 and iterate for $k = 0, 1, \dots$ until $\mu_{k+1} = \mu_k$:

  1. *Policy evaluation:* Find *average and differential costs* $J_k \in \mathbb{R}$ and $h_k \in \mathbb{R}^m$ satisfying for $i = 1, \dots, m$:

     $$J_k + h_k(i) = \mathbb{E}[g(i, \mu_k(i), W)] + \sum_{j=1}^{m} P_{\mu_k, ij} h_k(j).$$

     $(J_k + h_k(i) = \mathbb{T}_{\text{avg},\mu_k} \mathbf{h}_k)$

  2. *Policy improvement:* Find new policy μ_{k+1} satisfying for $i = 1, \dots, m$:

     $$\mu_{k+1}(i) + \sum_{j=1}^{m} P_{\mu_{k+1}, ij} h_k(j) = \min_{u \in \mathcal{U}} \left[\mathbb{E}_W[g(i, u, W)] + \sum_{j=1}^{m} P_{u,ij} h_k(j)\right].$$

     $(\mathbb{T}_{\text{avg},\mu_{k+1}} \mathbf{h}_k = \mathbb{T}_{\text{avg}} \mathbf{h}_k)$

Average Infinite-Cost Case with L Cost-Constraints

- Optimal average infinite horizon cost:

$$\bar{J}^*(p_0) := \min_{\pi} \bar{J}^{\pi}(p_0)$$

where minimum is only over policies π satisfying

$$\overline{\lim_{N\to\infty}} \frac{1}{N} \mathbb{E}_{X_0,\{W_k\}}\left[\sum_{k=0}^{N-1} d_\ell(X_k, \mu_k(X_k), W_k)\right] \leq D_\ell, \qquad \ell = 1, \ldots, L.$$

- Similar to before we can prove that we can restrict to stationary policies where the limsups are proper limits.

- Can express the average cost and the constraints with the state-action frequencies $\nu_\infty^\mu(i,u)$ of the stationary policies μ

Linear Programme for Optimal Average Cost with Constraints

"Dual Problem" for Average Costs and Constraints

$$\bar{J}^* = \min_{\nu(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W[g(i,u,W)] \cdot \nu(i,u)$$

subject to:

$$\sum_{v} \nu(i,v) = \sum_{j=1}^{m} \sum_{u} P_{u,ij} \cdot \nu(j,u), \qquad i = 1, \ldots, m,$$

$$\sum_{i,u} \nu(i,u) = 1,$$

$$\sum_{i=1}^{m} \sum_{u} \mathbb{E}_W[d_\ell(i,u,W)] \cdot \nu(i,u) \leq D_\ell, \qquad \ell = 1, \ldots, L.$$

- Optimal $\rho^*(i,u) > 0$ for at most $m + L$ pairs (i,u) (since there are $m + L$ lin. ind. constraints)

Maybe randomized actions in optimal policy $\mu^* = \frac{\nu^*(i,u)}{\sum_v \nu^*(i,v)}$

Optimal Policy has L Randomization Points

- Randomized stationary policies with L randomization points optimal
- Consider $L = 1$ and optimal ν^* with $m + 1$ positive entries:

$$\nu^*(1, u_1), \nu^*(2, u_2), \nu^*(3, u_3), \ldots, \nu^*(m, u_m) > 0$$

  and for some $j \in \{1, \ldots, m\}$ and $u_j' \neq u_j$:

$$\nu^*(j, u_j') > 0.$$

  All other entries $\nu^*(i, u) = 0$.

Initial Randomization Suffices

- Idea: Randomize only at the beginning!

- Create the m-ary state-action frequencies

$$\nu_1(i,u) = \begin{cases} \nu^*(j,u_j) + \nu^*(j,u_j') & i=j, u=u_j \\ 0 & i=j, u=u_j' \\ \mu^*(i,u), & \text{otherwise.} \end{cases}$$

$$\nu_2(i,u) = \begin{cases} 0 & i=j, u=u_j \\ \nu^*(j,u_j) + \nu^*(j,u_j') & i=j, u=u_j' \\ \mu^*(i,u), & \text{otherwise.} \end{cases}$$

- Construct the deterministic stationary policies

$$\mu_1(u|i) = \frac{\nu_1(i,u)}{\sum_v \nu_1(i,v)} \qquad \mu_2(u|i) = \frac{\nu_2(i,u)}{\sum_v \nu_2(i,v)}$$

- At the beginning play each *deterministic* policy μ_l with prob. q_l, $l=1,2$,

$$q_1 := \frac{\nu^*(j,u)}{\nu^*(j,u_j) + \nu^*(j,u_j')} \qquad q_2 := \frac{\nu^*(j,u')}{\nu^*(j,u_j) + \nu^*(j,u_j')}$$

Initial Randomization Suffices, continued

- The expected cost of this *mixed strategy* is:

$$
\begin{aligned}
q_1 \bar{J}^{\mu_1} + q_2 \bar{J}^{\mu_2} &= q_1 \sum_{i,u} \mathbb{E}[g(i,u,W)]\nu_\infty^{\mu_1}(i,u) + q_2 \sum_{i,u} \mathbb{E}[g(i,u,W)]\nu_\infty^{\mu_2}(i,u) \\
&= q_1 \sum_{i,u} \mathbb{E}[g(i,u,W)]\nu_1(i,u) + q_2 \sum_{i,u} \mathbb{E}[g(i,u,W)]\nu_2(i,u) \\
&= \sum_{i,u} \mathbb{E}[g(i,u,W)]\left(q_1 \cdot \nu_1(i,u) + q_2 \cdot \nu_2(i,u)\right) \\
&= \sum_{i,u} \mathbb{E}[g(i,u,W)]\nu^*(i,u) = \bar{J}^*
\end{aligned}
$$

- The mixed strategy also satisfies the constraints for each $\ell = 1, \ldots, L$:

$$
\begin{aligned}
q_1 \sum_{i,u} \mathbb{E}[d_\ell(i,u,W)]\nu_1(i,u) &+ q_2 \sum_{i,u} \mathbb{E}[d_\ell(i,u,W)]\nu_2(i,u) \\
&= \sum_{i,u} \mathbb{E}[d_\ell(i,u,W)]\left(q_1 \cdot \nu_1(i,u) + q_2 \cdot \nu_2(i,u)\right) \\
&= \sum_{i,u} \mathbb{E}[d_\ell(i,u,W)]\nu^*(i,u) \leq D_l l
\end{aligned}
$$

Optimal strategy: Randomly play one of L deterministic policies

Average Infinite-Cost Case with Constraints and Lagrange Multipliers

"Dual Problem" for Average Costs and Constraints with Lagrange Multipliers

$$\bar{J}^* = \sup_{\lambda_1,\ldots,\lambda_L \geq 0} \min_{\nu(i,u)\geq 0} \sum_{i=1}^{m} \sum_{u} \mathbb{E}_W \left[g(i,u,W) + \sum_{\ell} \lambda_\ell d_\ell(i,u,W) \right] \cdot \nu(i,u) - \sum_{\ell=1}^{L} \lambda_\ell D_\ell$$

subject to:

$$\sum_{v} \nu(i,v) = \sum_{j=1}^{m} \sum_{u} P_{u,ij} \cdot \nu(j,u) \qquad i = 1,\ldots,m,$$

$$\sum_{i,u} \nu(i,u) = 1.$$

- For each $\lambda_1, \ldots, \lambda_L$ a deterministic policy μ is optimal.

Sequential Decision Processes, Master MICAS, Part I

Michèle Wigger

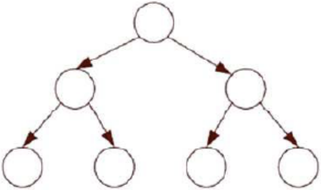
Telecom Paris, 8 January 2021

Lecture 7 – Algorithmic Dynamic Programming

Algorithmic Paradigms

- Greedy Algorithm
  - Construct solution incrementally
  - Greedily choose the “right” subproblem by optimizing a local criterion

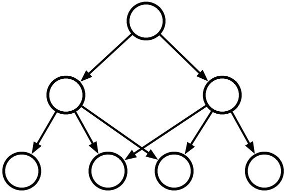
- Divide and Conquer
  - Divide a problem into *non-overlapping* subproblems
  - Solve each subproblem (in any order)
  - Combine solutions of subproblems to obtain solution to initial problem
  - Top-down approach



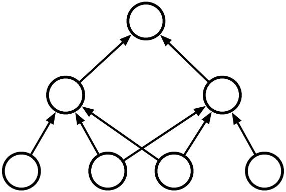
Dynamic Programming (Bellman) Principle

- Breaking the problem into *overlaping* subproblems
- Calculate and store optimal solutions to subproblems
- Combine solutions to subproblems to solve the initial problem
- Solutions can be cached (stored) and reused

Top-down: *Memoization*



Bottom-up: *Tabulation*



Example: Binomial Coefficient $C_n^k = \binom{n}{k} = \frac{n!}{k!(n-k)!}$

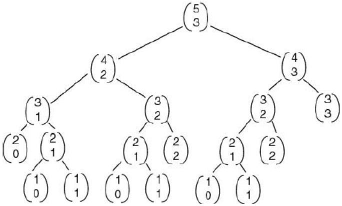
Recursive formula:

$$C_n^k = \begin{cases} \binom{n-1}{k-1} + \binom{n-1}{k} & 0 < k < n \\ 1 & \text{otherwise} \end{cases}$$

Divide and Conquer Approach:

Function C(n, k)

1. if $(k = 0)$ or $(k = n)$ return 1;
2. else return $C(n-1, k-1) + C(n-1, k)$;



- Time complexity:
  - Exponential number of recursive calls: $O\left(\binom{n}{k}\right) \approx 2\binom{n}{k}$

Example: Binomial Coefficient, continued

Pascal-triangle approach: Dynamic Programming with memoization based on 2-dimensional table

Function C-mem(n, k)

1. for $(i = 0; i \leq n; i++)$
2. for $(j = 0; j \leq \min(i, k); j++)$
3. if $(i = 0)$ or $(j = i)$, $T[i][j] = 1;$
4. else $T[i][j] = T[i-1][j-1] + T[i-1][j];$
5. return $T[n][k];$

	0	1	2	3	$\cdots$	$n-1$	n
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\ddots$		
$n-1$	1	$n-1$	$\binom{n-1}{2}$	$\binom{n-1}{3}$	$\cdots$	1	
n	1	n	$\binom{n}{2}$	$\binom{n}{3}$	$\cdots$	n	1

- Top -Down Approach

- Auxiliary space $O(nk)$ and time-complexity $O(nk)$.

Example: Binomial Coefficient (3)

- Dynamic programming solution: Tabulation
- Create table with 1 dimension to compute small numbers
- Compute next row of pascal triangle using previous row
  Function C-dyn(n, k)

  1. $T[0] = 1$;
  2. for $(i = 0; i \leq n; i++)$
  3.   for $(j = \min(i, k); j > 0; j--)$ do $T[j] = T[j] + T[j-1]$;
  4. return $T[k]$;

- Time complexity:
  - Table of k elements $\Rightarrow$ Auxiliary space $O(k)$
  - Time complexity: $O(nk)$
- Optimized-space bottom-up DP approach

How to design Dynamic Programming Solution

- Define subproblems
- Identify recursive relation between subproblems
- Avoid similar computation
- Resolve original problem by combining solutions of subproblems
- Tabulation approach:
  - Recognize and solve the base cases
  - Deduce dynamic programming algorithm in a bottom-up way
- Memoization approach:
  - Deduce dynamic programming algorithm in a top-down way

Sequential Decision Processes, Master MICAS, Part I

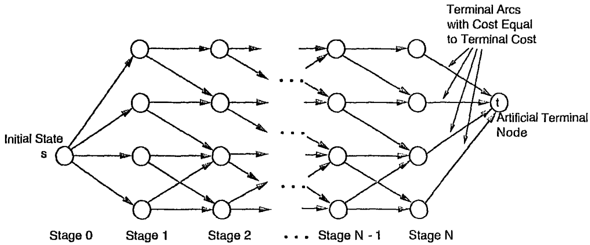
Michèle Wigger

Telecom Paris, 8 January 2021

Lecture 7 – Some Shortest Paths Algorithms

Deterministic MDPs and Shortest-Path Problems

- No disturbance $\rightarrow$ state evolution $x_{k+1} = f(x_k, u_k)$ and cost $g_k(x_k, u_k)$
- Graph representation:

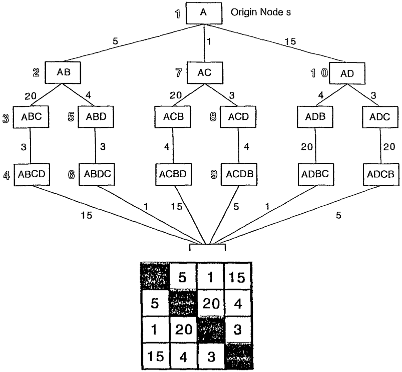


  - At each stage $k = 1, 2, \ldots, N$ there is a node for each $x_k \in \mathcal{X}$
  - Arrows indicate transitions for different actions $\rightarrow$ label arrows with actions u_k and costs $g_k(x_k, u_k)$

- Total cost $J_{0\rightarrow N,\pi}$ is the sum of the costs on the path indicated by π

Finding minimum total cost $J_{0\rightarrow N,\pi}$ equivalent to finding "shortest path"
$\rightarrow$ DP algorithm can be run in reverse order

Travelling Salesman Problem and Label Correcting Method



Initialize $d_s = 0$ and
$d_2 = \cdots = d_t = \text{upper} = \infty$

Label Correcting Algorithm

Step 1: Remove a node i from OPEN and for each child j of i, execute step 2.

Step 2: If $d_i + a_{ij} < \min\{d_j, \text{UPPER}\}$, set $d_j = d_i + a_{ij}$ and set i to be the parent of j. In addition, if $j \neq t$, place j in OPEN if it is not already in OPEN, while if $j = t$, set UPPER to the new value $d_i + a_{it}$ of d_t.

Step 3: If OPEN is empty, terminate; else go to step 1.

Iter. No.	Node Exiting OPEN	OPEN at the End of Iteration	UPPER
0	-	1	∞
1	1	2, 7,10	∞
2	2	3, 5, 7, 10	∞
3	3	4, 5, 7, 10	∞
4	4	5, 7, 10	43
5	5	6, 7, 10	43
6	6	7, 10	13
7	7	8, 10	13
8	8	9, 10	13
9	9	10	13
10	10	Empty	13

- State space depends on stage k

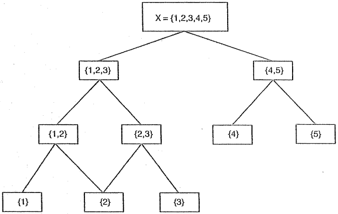
- Dijkstra's method always chooses the node in OPEN with smallest d_i.
- Bellman-Ford algorithm chooses the node in OPEN as first-in first-out.

The Branch-and-Bound Algorithm

- Wish to minimize cost function $f(\cdot)$ over all elements of $\mathcal{X}$

Find functions $\overline{f}$ and $\underline{f}$ over subsets $\mathcal{Y} \subseteq \mathcal{X}$ such that :

$$\underline{f}(\mathcal{Y}) \leq \min_{x \in \mathcal{Y}} f(x) \leq \overline{f}(\mathcal{Y}), \qquad \forall \mathcal{Y} \subseteq \mathcal{X}.$$



- Construct a tree with subsets of $\mathcal{X}$ $\rightarrow$ including *all* singletons!
- If $\mathcal{Y}_i \subseteq \mathcal{Y} \Rightarrow \mathcal{Y}$ is a parent of $\mathcal{Y}_i$
- Label branch from $\mathcal{Y}$ to $\mathcal{Y}_i$ by $\underline{f}(\mathcal{Y}_i) - \underline{f}(\mathcal{Y}) \Rightarrow$ path length from $\mathcal{X}$ to $\mathcal{Y}$ equals $\underline{f}(\mathcal{Y})$

Branch-and-Bound Algorithm

Step 1: Remove a node Y from OPEN. For each child Y_j of Y, do the following: If $\underline{f}_{Y_j} <$ UPPER, then place Y_j in OPEN. If in addition $\overline{f}_{Y_j} <$ UPPER, then set UPPER $= \overline{f}_{Y_j}$, and if Y_j consists of a single solution, mark that solution as being the best solution found so far.

Step 2: **(Termination Test)** If OPEN is nonempty, go to step 1. Otherwise, terminate; the best solution found so far is optimal.