Stochastic Models and Optimization: Problem Set 1

Roger Garriga Calleja, José Fernando Moreno Gutiérrez, David Rosenfeld, Katrina Walker

February 27, 2017

Problem 3 (Multiplicative cost): *In the framework of the basic problem, consider the case where the cost has the multiplicative form*

$$\mathbb{E}_{\{w_k\}}\left\{g_N(x_N)\cdot g_{N-1}(x_{N-1},u_{N-1},w_{N-1})\cdots g_0(x_0,u_0,w_0)\right\}$$

Develop a DP-like algorithm for this problem assuming that $g_k(x_k,u_k,w_k)>0$, for all x_k,u_k,w_k and k.

<u>Primitives:</u>

State: The state of the system at the beginning of the period k is x_k.

Control: The control or decision at period k is u_k.

Uncertainty: The uncertainty at period k is w_k.

Dynamics: The dynamics of the problem at period k is $x_{k+1}=f_k(x_k,u_k,w_k)$, for a certain f_k.

Cost: The cost at period k is $f_k(x_k,u_k,w_k)$ and has a multiplicative form (so the cost from the period 0 to the period $k\neq N$ will be $\prod_{i=0}^{k} g_i(x_i,u_i,w_i)$ and $g_N(x_N)\prod_{i=0}^{k-1} g_i(x_i,u_i,w_i)$ if $k=N$).

As in the DP problem, we can take $J_k(x_k)$ as the cost-to-go of the $N-k$ element. Then,

$$J_N(x_N)=g(x_N).$$

And from there on considering U_k the set of the possible decisions,

$$J_k(x_k)=\min_{u_k\in U_k}\mathbb{E}_{w_k}\left\{g_k(x_k,u_k,w_k)J_{k+1}(f_k(x_k,u_k,w_k))\right\}.$$

Let's prove that this makes sense, meaning that if the minimum of the expected cost-to-go in the $(k+1)$th state is $J_{k+1}(x_{k+1})$ then the minimum of the expected cost-to-go in the kth state will be $J_k(x_k)$ as we have defined it. To do so, we will use induction.

Let us call $J^*(x_k)$ the real minimum of the expected cost-to-go in the kth state. It is clear that $J_N^*(x_N)=J_N(x_N)=g_N(x_N)$ (because it does not depend on any w). Now, assume that $J^*_{k+1}(x_{k+1})=J_{k+1}(x_{k+1})$. Then, considering $J_k^*(x_k)$,

$$J_k^*(x_k)=\min_{u_k\dots,u_{N-1}}\mathbb{E}_{w_k,\dots,w_{N-1}}\left\{g_k(x_k,u_k,w_k)\prod_{i=k+1}^{N-1}\left(g_i(x_i,u_i,w_i)\right)g_N(x_N)\right\}.$$

Since $g_k(x_k,u_k,w_k)$ only depends on period k we can put it out of the expectation and the minimum over the subsequent variables ($u_{k+1},\dots,u_{N-1}$ and $w_{k+1},\dots,w_{N-1}$). So we get

$$\begin{aligned}J_k^*(x_k)&=\min_{u_k}\mathbb{E}_{w_k}\left\{g_k(x_k,u_k,w_k)\min_{u_{k+1},\dots,u_{N-1}}\mathbb{E}_{w_{k+1},\dots,w_{N-1}}\left\{\prod_{i=k+1}^{N-1}\left(g_i(x_i,u_i,w_i)\right)g_N(x_N)\right\}\right\}=\\&=\min_{u_k}\mathbb{E}_{w_k}\left\{g_k(x_k,u_k,w_k)J^*_{k+1}(x_{k+1})\right\},\end{aligned}$$

since $J_{k+1}(x_{k+1}) = J^*_{k+1}(x_{k+1}) = \min\limits_{u_{k+1},\ldots,u_{N-1}} \mathbb{E}_{w_{k+1},\ldots,w_{N-1}} \left\{ \prod\limits_{i=k+1}^{N-1} (g_i(x_i, u_i, w_i))\, g_N(x_N) \right\}$ by hypothesis of induction,

$$J^*_k(x_k) = \min_{u_k} \mathbb{E}_{w_k} \{g_k(x_k, u_k, w_k) J_{k+1}(x_{k+1})\}.$$

Q.E.D.

Problem 5 (Traveling Repairman Problem): *A repairman must service N sites, which are located along a line and are sequentially numbered 1, 2, . . . , N. The repairman starts at a given site s with 1 < s < N, and is constrained to service only sites that are adjacent to the ones serviced so far, i.e. if he has already serviced sites i, i+1,...j, then he may service only site i-1 (assuming 1 <i) or that site j+1 (assuming j <N). There is a waiting cost of c_i for each time period that site i has remained unserviced and there is a travel cost t_{ij} for servicing site j right after site i. Reformulate the problem within the DP framework.*

Primitives:
S: the origin at period k
T: the destination
α_{ij}: $t_{ij} + \sum_{j=k+1}^{N} c_j$
d_i : path of minimum total cost from s to i
P_i : parent of node
Open: set of nodes whose labels may need correction
Upper: lowest costs from s - t

Initialization:

$$d_S = \sum_{j=1}^{N} c_j$$
$$d_i = \inf$$
$$\forall\ j \neq S$$
$$\text{Open} = S$$
$$\text{Upper} = \infty$$

Algorithm:

$$1) \text{ Remove } i \in \text{Open}$$
$$\forall\ j \text{ child of i execute (2)}$$
$$2) \text{ if } d_i + \alpha_{ij} < \min\ \{d_j, \text{Upper}\}$$
$$\text{then } d_j = d_i + \alpha_{ij}$$
$$P_j = i$$
$$\text{If } j \neq t, \text{put } j \text{ in Open}$$
$$\text{If } j = t, \text{Upper } = d_i + \alpha_{ij}$$
$$3) \text{ If Open} \neq 0 \text{terminate, else to go to (1)}$$