14D006 Stochastic Models and Optimization

Problemset 1

Daniel Bestard, Michael Cameron, Hans-Peter Höllwirth, Akhil Lohia

March 2, 2017

1 Machine Maintenance

The first step is to identify the variables of the DP algorithm for this specific problem:

- x_k: state of the system at period k. The possible states of the machine are $x_k \in \{Run, Broken\} = \{R, B\}$
- u_k: decision variables of the system at period k. In this problem depending on the state of the machine the decision variable changes. That is, if the machine is running, then we could apply maintenance or not, $u_k(R) \in \{\text{maintenance, not maintenance}\} = \{m, n\}$. On the other hand, if the machine is broken, then we could either repair it or replace it, $u_k(B) \in \{repair, replace\} = \{r, l\}$.
- w_k: uncertainty of the system at period k. In this problem the uncertainty comes from the fact that we do not know whether the machine will break or not. Therefore, the values of the variable that represents the uncertainty of the system are the possible states of the system at each period. That is, $w_k = x_k \in \{R, B\}$.

Once the variables of the problem have been defined the next step when applying the DP algorithm is to find the expression that explains the dynamics of this problem. As explained in the previous paragraph, the state of the machine in the next period is fully specified by the variable that contains the uncertainty of the system. That is,

$$x_{k+1} = f_k(x_k, u_k, w_k) = w_k(x_k, u_k)$$

The previous expression is one of the components of the objective function of the DP algorithm. The other expression to be defined is the one that specifies the gains of each possible situation. This function is:

$$g_k(x_k, u_k, w_k) = \begin{cases} -20 & if\, u_k = m, w_k = B \\ 80 & if\, u_k = m, w_k = R \\ 0 & if\, u_k = n, w_k = B \\ 100 & if\, u_k = n, w_k = R \\ -40 & if\, u_k = r, w_k = B \\ 60 & if\, u_k = r, w_k = R \\ 10 & if\, u_k = l, w_k = R \end{cases}$$

Note that in the previous function there is one situation that has been omitted, which is when the machine is replaced and breaks, $u_k = l, w_k = B$. The reason why this option is not included is because it is not possible by construction. As the exercise specifies, if the machine is replaced then it is guaranteed to work for the whole week.

The last part to be defined before applying the DP algorithm is the transition probabilities, $p(w_k|x_k, u_k)$

$$\begin{aligned} P(R|R,m) = 0.6 \quad P(B|R,m) = 0.4 \\ P(R|R,n) = 0.3 \quad P(B|R,n) = 0.7 \\ P(R|B,r) = 0.6 \quad P(B|B,r) = 0.3 \\ P(R|B,l) = 1.0 \quad P(B|B,l) = 0.0 \end{aligned}$$

Now we are ready to apply the DP algorithm, which has the following shape:

$$\begin{aligned} J_N(x_N) &= g_N(x_k) \\ J_k(x_k) &= \max_{u_k \in U_k(x_k)} {}_{w_k} \left[g_k(x_k, u_k, w_k) + J_{k+1}[f_k(x_k, u_k, w_k)] \right] \end{aligned}$$

Let's apply the DP algorithm:

$$
\begin{aligned}
J_3(R) &= max\Big\{ {}_{w_k}\big[g_3(R,m,w_k) + J_4(w_k(R,m))\big],\ {}_{w_k}\big[g_3(R,n,w_k) + J_4(w_k(R,n))\big]\Big\} \\
&= max\Big\{ \big[g_3(R,m,R) + J_4(R)\big]P(R|R,m) + \big[g_3(R,m,B) + J_4(B)\big]P(B|R,m), \\
&\qquad\qquad \big[g_3(R,n,R) + J_4(R)\big]P(R|R,n) + \big[g_3(R,n,B) + J_4(B)\big]P(B|R,n)\Big\} \\
&= max\big\{(80+0)0.6 + (-20+0)0.4, (100+0)0.3 + (0+0)0.7\big\} \\
&= max\big\{40, 30\big\} \\
&= 40
\end{aligned}
$$

$$\boxed{\mu_3(R) = m}$$

$$
\begin{aligned}
J_3(B) &= max\Big\{ {}_{w_k}\big[g_3(B,r,w_k) + J_4(w_k(B,r))\big],\ {}_{w_k}\big[g_3(B,l,w_k) + J_4(w_k(B,l))\big]\Big\} \\
&= max\Big\{ \big[g_3(B,r,R) + J_4(R)\big]P(R|B,r) + \big[g_3(B,r,B) + J_4(B)\big]P(B|B,r), \\
&\qquad\qquad \big[g_3(B,l,R) + J_4(R)\big]P(R|B,n)\Big\} \\
&= max\big\{(60+0)0.6 + (-40+0)0.4, (10+0)1\big\} \\
&= max\big\{20, 10\big\} \\
&= 20
\end{aligned}
$$

$$\boxed{\mu_3(B) = r}$$

$$
\begin{aligned}
J_2(R) &= max\Big\{ {}_{w_k}\big[g_2(R,m,w_k) + J_3(w_k(R,m))\big],\ {}_{w_k}\big[g_2(R,n,w_k) + J_3(w_k(R,n))\big]\Big\} \\
&= max\Big\{ \big[g_2(R,m,R) + J_3(R)\big]P(R|R,m) + \big[g_2(R,m,B) + J_3(B)\big]P(B|R,m), \\
&\qquad\qquad \big[g_2(R,n,R) + J_3(R)\big]P(R|R,n) + \big[g_2(R,n,B) + J_3(B)\big]P(B|R,n)\Big\} \\
&= max\big\{(80+40)0.6 + (-20+20)0.4, (100+40)0.3 + (0+20)0.7\big\} \\
&= max\big\{72, 56\big\} \\
&= 72
\end{aligned}
$$

$$\boxed{\mu_2(R) = m}$$

$$
\begin{aligned}
J_2(B) &= max\Big\{ {}_{w_k}\big[g_2(B,r,w_k) + J_3(w_k(B,r))\big],\ {}_{w_k}\big[g_2(B,l,w_k) + J_3(w_k(B,l))\big]\Big\} \\
&= max\Big\{ \big[g_2(B,r,R) + J_3(R)\big]P(R|B,r) + \big[g_2(B,r,B) + J_3(B)\big]P(B|B,r), \\
&\qquad\qquad \big[g_2(B,l,R) + J_3(R)\big]P(R|B,n)\Big\} \\
&= max\big\{(60+40)0.6 + (-40+20)0.4, (10+40)1\big\} \\
&= max\big\{52, 50\big\} \\
&= 52
\end{aligned}
$$

$$\boxed{\mu_2(B) = r}$$

$$
\begin{aligned}
J_1(R) &= max\Big\{ {}_{w_k}\big[g_1(R,m,w_k) + J_2(w_k(R,m))\big],\ {}_{w_k}\big[g_1(R,n,w_k) + J_2(w_k(R,n))\big]\Big\} \\
&= max\Big\{ \big[g_1(R,m,R) + J_2(R)\big]P(R|R,m) + \big[g_1(R,m,B) + J_2(B)\big]P(B|R,m), \\
&\qquad \big[g_1(R,n,R) + J_2(R)\big]P(R|R,n) + \big[g_1(R,n,B) + J_2(B)\big]P(B|R,n)\Big\} \\
&= max\big\{(80+72)0.6 + (-20+52)0.4, (100+72)0.3 + (0+52)0.7\big\} \\
&= max\big\{104, 88\big\} \\
&= 104
\end{aligned}
$$

$$\boxed{\mu_1(R) = m}$$

$$
\begin{aligned}
J_1(B) &= max\Big\{ {}_{w_k}\big[g_1(B,r,w_k) + J_2(w_k(B,r))\big],\ {}_{w_k}\big[g_1(B,l,w_k) + J_2(w_k(B,l))\big]\Big\} \\
&= max\Big\{ \big[g_1(B,r,R) + J_2(R)\big]P(R|B,r) + \big[g_1(B,r,B) + J_2(B)\big]P(B|B,r), \\
&\qquad \big[g_1(B,l,R) + J_2(R)\big]P(R|B,n)\Big\} \\
&= max\big\{(60+72)0.6 + (-40+52)0.4, (10+72)1\big\} \\
&= max\big\{84, 82\big\} \\
&= 84
\end{aligned}
$$

$$\boxed{\mu_1(B) = r}$$

$$
\begin{aligned}
J_0(R) &= g_0(R,n,R) + J_1(R) \\
&= 100 + 104 \\
&= 204
\end{aligned}
$$

$$\boxed{\mu_0(R) = n}$$

The optimal solution at each period for each possible state of the system can be found in the previous boxes. Note that $J_0(R)$ has no expectation because there is no uncertainty

given that the problem specifies that at the beginning there is a new machine that is guaranteed not to break during the first week. For the same reasoning, it does not make sense to compute $J_0(B)$.

2 Discounted Cost

In the framework of the basic problem, consider the case where the cost is of the form

$$\mathop{\mathbb{E}}_{\{w_k\}}[\alpha^N g_N(x_N) + \sum_{k=0}^{N-1} \alpha^k g_k(x_k, u_k, w_k)]$$

where $\alpha \in (0, 1)$ is a discount factor.

Let J_k^* be the optimal value of the $(N-k)$-tail problem with cost function $g_N^*(x_N) = \alpha^N g_N(x_N)$ and $g_k^*(x_k) = \alpha^k g_k(x_k, u_k, w_k)$. Then we have

$$\begin{aligned} J_N^*(x_N) &= g_N^*(x_N) \\ &= \alpha^N g_N(x_N) \\ \alpha^{-N} J_N^*(x_N) &= g_N(x_N) \end{aligned}$$

$$\begin{aligned} J_k^*(x_k) &= \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k}\left[g_k^*(x_k, u_k, w_k) + J_{k+1}^*(f_k(x_k, u_k, w_k))\right] \\ &= \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k}\left[\alpha^k g_k(x_k, u_k, w_k) + J_{k+1}^*(f_k(x_k, u_k, w_k))\right] \\ \alpha^{-k} J_k^*(x_k) &= \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k}\left[g_k(x_k, u_k, w_k) + \alpha^{-k} J_{k+1}^*(f_k(x_k, u_k, w_k))\right] \\ &= \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k}\left[g_k(x_k, u_k, w_k) + \alpha\alpha^{-(k+1)} J_{k+1}^*(f_k(x_k, u_k, w_k))\right] \end{aligned} \tag{1}$$

Now let $J_k(x_k) = \alpha^{-k} J^*(x_k)$ and so we get the DP-like algorithm

$$\begin{aligned} J_N(x_N) &= \alpha^{-N} J^*(x_N) \\ &= g_N(x_N) \end{aligned}$$

$$\begin{aligned} J_k(x_k) &= \alpha^{-k} J^*(x_k) \\ &= \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k}\left[g_k(x_k, u_k, w_k) + \alpha\alpha^{-(k+1)} J_{k+1}^*(f_k(x_k, u_k, w_k))\right] \\ &= \min_{u_k \in U_k(x_k)} \mathop{\mathbb{E}}_{w_k}\left[g_k(x_k, u_k, w_k) + \alpha J_{k+1}(f_k(x_k, u_k, w_k))\right] \end{aligned} \tag{2}$$

3 Multiplicative Cost

In the framework of the basic problem, consider the case where the cost has the multiplicative form

$$_{\{w_k\}}[g_N(x_N)g_{N-1}(x_{N-1},u_{N-1},w_{N-1})...g_0(x_0,u_0,w_0)]$$

assuming that $g_k(x_k,u_k,w_k)>0$, for all x_k, u_k, w_k, and k.

4 Knapsack Problem

Given the total capacity of the vessel z, consider adding different quantities of the N items sequentially. Hence, at each stage (item) of the problem, the decision variable is the quantity of that item to be loaded on the cargo. Define z_i as the remaining capacity of the vessel before loading item i. Hence, $z_0=z$. Thus, z_i can be considered as the state of the system at each stage. Define x_i to be the quantity of i^{th} item loaded on the vessel. Hence x_i is the control variable in our formulation. As given, w_i is the weight of this item type. Then we can give the following formulation of our dynamic programming framework:

Dynamics: $z_{i+1}=z_i-x_iw_i$
Reward: $g_i(z_i,x_i)=x_iv_i$
Constraints: $X_i(z_i)=\left\{x_i\in{}_+ : x_i\leq\frac{z_i}{w_i}\right\}$

We have the terminal condition as follows:

$$J_N(z_N)=\lfloor z_N/w_N\rfloor$$

Then we can use the regular DP recursion:

$$J_i(z_i)=\max_{x_i\in X_i}\{x_iv_i+J_{i+1}(z_i-x_iw_i)\}$$

Using this, we can calculate J_0 which will be the most valuable cargo for the vessel with capacity z.

5 Traveling Repairman Problem

Denote the location of the repairman at the beginning of stage k with x_k. Thus, x_1 will be s, which is the starting location where $1<s<N$. Let a_k and b_k denote the first and last site respectively which have already been serviced at the beginning of period k. Since the repairman only services adjacent sites, $1\leq a_k\leq b_k\leq N$. Thus, x_k,a_k, and b_k together specify the state of the system. The decision variable u_k is the site that will be serviced next. Hence, $x_{k+1}=u_k$.

At each stage, the set of serviced sites increases to include the site serviced in that stage, i.e., either $a_{k+1} = a_k - 1$, or $b_{k+1} = b_k + 1$.

The cost of servicing includes the waiting cost c_i for the unserviced sites i and the travel cost from the current site to the next site. Thus, cost: $g_k(x_k, a_k, b_k, u_k) = \sum_{i<a_k, i>b_k} c_i + t_{x_k u_k}$.

The constraints in the problem are defined by the sites that can be covered by the repairman given the sites already serviced.

$$U_k(a_k, b_k) = \begin{cases} \{a_k - 1, b_k + 1\} & if\, 1 < a_k, b_k < N \\ b_k + 1 & if\, a_k = 1, b_k < N \\ a_k - 1 & if\, 1 < a_k, b_k = N \end{cases}$$

Now we can use the usual DP algorithm. We have the following terminal condition (where x_{N+1} can be replaced by x_0):

$$J_N(x_{N+1}, a_k = 1, b_k = N) = 0$$

Hence, we use:

$$J_k(x_k, a_k, b_k) = \min_{u_k \in U_k(a_k, b_k)} \left\{ \sum_{i<a_k, i>b_k} c_i + t_{x_k u_k} + J_{k+1}(u_k, \{a_k, b_k\} \cup \{u_k\}) \right\}$$

This expression can be used to find the required minimum cost path $J_1(s)$.