Dynamic Programming and Optimal Control

SECOND EDITION

Dimitri P. Bertsekas

Massachusetts Institute of Technology

Selected Theoretical Problem Solutions

Last Updated 3/25/2006

Athena Scientific, Belmont, Mass.

WWW site for book information and orders

http://www.athenasc.com/

NOTE

This solution set is meant to be a significant extension of the scope and coverage of the book. It includes solutions to all of the book's exercises marked with the symbol www .

The solutions are continuously updated and improved, and additional material, including new problems and their solutions are being added. Please send comments, and suggestions for additions and improvements to the author at **dimitrib@mit.edu**

The solutions may be reproduced and distributed for personal or educational uses.

Solutions Vol. I, Chapter 1

1.5 www

We first show the result given in the hint. We have for all $\mu \in M$

$$\max_{w\in W}\Big\{G_0(w)+G_1\big[\,f(w),\mu(f(w))\,\big]\Big\}\geq \max_{w\in W}\Big\{G_0(w)+\min_{u\in U}G_1\big[f(w),u\big]\Big\}$$

and, taking min over $\mu \in M$ we obtain

$$\min_{\mu\in M}\max_{w\in W}\Big\{G_0(w)+G_1\big[\,f(w),\mu(f(w))\,\big]\Big\}\geq \max_{w\in W}\Big\{G_0(w)+\min_{u\in U}G_1\big[f(w),u\big]\Big\}. \tag{1}$$

We must therefore show the reverse inequality. For any $\epsilon > 0$, let $\mu_\epsilon \in M$ be such that

$$G_1\big[\,f(w),\mu_\epsilon(f(w))\,\big]\leq \min_{u\in U}G_1\big[f(w),u\big]+\epsilon, \qquad \forall\, w\in W$$

(Such a μ_ϵ exists because of the assumption that $\min_{u\in U}G_1\big[f(w),u\big] > -\infty$.) Then

$$\begin{aligned}\min_{\mu\in M}\max_{w\in W}\Big\{G_0(w)+G_1\big[\,f(w),\mu(f(w))\,\big]\Big\}&\leq \max_{w\in W}\Big\{G_0(w)+G_1\big[\,f(w),\mu_\epsilon(f(w))\,\big]\Big\}\\ &\leq \max_{w\in W}\Big\{G_0(w)+\min_{u\in U}G_1\big[f(w),u\big]\Big\}+\epsilon\end{aligned}.$$

Since $\epsilon > 0$ can be taken arbitrarily small we obtain the reverse inequality to Eq. (1), and thus the desired result.

To see how this result can fail without the condition $\min_{u\in U}G_1\big[f(w),u\big] > -\infty$ for all w, let u be a real number, let $w = (w_1, w_2)$ be a two-dimensional vector, let there be no constraints on u and w ($U = \Re$, $W = \Re \times \Re$, where $\Re$ is the real line), let $G_0(w) = w_1$, $f(w) = w_2$, and $G_1[f(w), u] = f(w) + u$. Then

$$\max_{w\in W}\Big\{G_0(w)+G_1\big[\,f(w),\mu(f(w))\,\big]\Big\}=\max_{w_1\in\Re,\ w_2\in\Re}\big\{w_1+w_2+\mu(w_2)\big\}=\infty, \qquad \forall\ \mu\in M,$$

so that

$$\min_{\mu\in M}\max_{w\in W}\Big\{G_0(w)+G_1\big[\,f(w),\mu(f(w))\,\big]\Big\}=\infty.$$

On the other hand,

$$\max_{w\in W}\Big\{G_0(w)+\min_{u\in U}G_1\big[f(w),u\big]\Big\}=\max_{w_1\in\Re,\ w_2\in\Re}\Big\{w_1+\min_{u\in\Re}\{w_2+u\}\Big\}=-\infty,$$

since $\min_{u\in\Re}\{w_2 + u\} = -\infty$ for all w_2.

We now turn to showing the DP algorithm. We have

$$\begin{aligned}J^*(x_0)&=\min_{\mu_0}\cdots\min_{\mu_{N-1}}\max_{w_0\in W[x_0,\mu_0(x_0)]}\cdots\max_{w_{N-1}\in W[x_{N-1},\mu_{N-1}(x_{N-1})]}\left\{\sum_{k=0}^{N-1}g_k(x_k,\mu_k(x_k),w_k)+g_N(x_N)\right\}\\ &=\min_{\mu_0}\cdots\min_{\mu_{N-2}}\left\{\min_{\mu_{N-1}}\max_{w_0\in W[x_0,\mu_0(x_0)]}\cdots\max_{w_{N-2}\in W[x_{N-2},\mu_{N-2}(x_{N-2})]}\left\{\sum_{k=0}^{N-2}g_k(x_k,\mu_k(x_k),w_k)\right.\right.\\ &\left.\left.+\max_{w_{N-1}\in W[x_{N-1},\mu_{N-1}(x_{N-1})]}\big\{g_{N-1}(x_{N-1},\mu_{N-1}(x_{N-1}),w_{N-1})+J_N(x_N)\big\}\right\}\right\}\end{aligned}$$

Applying the result of the hint with the identifications

$$w = (w_0, w_1, \ldots w_{N-2}), \qquad u = u_{N-1}, \qquad f(w) = x_{N-1},$$

$$G_0(w) = \begin{cases} \sum_{k=0}^{N-2} g_k(x_k, \mu_k(x_k), w_k) & \text{if } w_k \in W_k[x_k, \mu_k(x_k)],\ \forall\ k, \\ \infty & \text{otherwise,} \end{cases}$$

$$G_1[f(w), u] = \begin{cases} \hat{G}_1[f(w), u] & \text{if } u \in U_{N-1}\big[f(w)\big], \\ \infty & \text{otherwise,} \end{cases}$$

where

$$\hat{G}_1[f(w), u] = \max_{w_{N-1} \in W_{N-1}[f(w), u]} \Big\{ g_{N-1}\big(f(w), u, w_{N-1}\big) + J_N\big(f_{N-1}(f(w), u, w_{N-1})\big) \Big\},$$

we have

$$J^*(x_0) = \min_{\mu_0} \cdots \min_{\mu_{N-2}}$$
$$\max_{w_0 \in W[x_0, \mu_0(x_0)]} \cdots \max_{w_{N-2} \in W[x_{N-2}, \mu_{N-2}(x_{N-2})]} \left\{ \sum_{k=0}^{N-2} g_k(x_k, \mu_k(x_k), w_k) + J_{N-1}(x_{N-1}) \right\}$$

The required condition $\min_{u \in U} G_1\big[f(w), u\big] > -\infty$ for all w is implied by the assumption $J_{N-1}(x_{N-1}) > -\infty$ for all x_{N-1}. Without this assumption, it can be seen that mathematical anomalies of the type demonstrated in the earlier example may arise.

By working with the preceding expression for $J^*(x_0)$ and by similarly continuing backwards, with $N-1$ in place of N, etc., after N steps we obtain $J^*(x_0) = J_0(x_0)$.

1.13 www

The DP algorithm is

$$J_N(x_N) = c_N' x_N$$

$$J_k(x_k) = \min_{u_k} \mathop{E}_{w_k, A_k} \left\{ c_k' x_k + g_k(u_k) + J_{k+1}\left[A_k x_k + f_k(u_k) + w_k\right] \right\}.$$

We will show that $J_k(x_k)$ is affine through induction. Clearly $J_N(x_N)$ is affine. Assume that $J_{k+1}(x_{k+1})$ is affine; that is,

$$J_{k+1}(x_{k+1}) = b_{k+1}' x_{k+1} + d_{k+1}.$$

Then

$$\begin{aligned} J_k(x_k) &= \min_{u_k} \mathop{E}_{w_k, A_k} \left\{ c_k' x_k + g_k(u_k) + b_{k+1}' A_k x_k + b_{k+1}' f_k(u_k) + b_{k+1}' w_k + d_{k+1} \right\} \\ &= \left[c_k' + b_{k+1}' \, E\left\{A_k\right\} \right] x_k + b_{k+1}' \, E\left\{w_k\right\} + \min_{u_k} \left\{ g_k(u_k) + b_{k+1}' f_k(u_k) \right\} + d_{k+1}. \end{aligned}$$

Note that $E\{A_k\}$ and $E\{w_k\}$ do not depend on x_k or u_k. If the optimal value is finite then $\min\{g_k(u_k) + b_{k+1}' f_k(u_k)\}$ is a real number and $J_k(x_k)$ is affine. Furthermore, if there is an optimal policy, for each k, the optimal control solves this minimization which is independent of x_k. Thus, there is an optimal policy that consists of constant functions.

1.16 www

(a) Given a sequence of matrix multiplications

$$M_1 M_2 \cdots M_k M_{k+1} \cdots M_N$$

we represent it by a sequence of numbers $\{n_1, \ldots, n_{N+1}\}$, where $n_k \times n_{k+1}$ is the dimension of M_k. Let the initial state be $x_0 = \{n_1, \ldots, n_{N+1}\}$. Then choosing the first multiplication to be carried out corresponds to choosing an element from the set $x_0 - \{n_1, n_{N+1}\}$. For instance, choosing n_2 corresponds to multiplying M_1 and M_2, which results in a matrix of dimension $n_1 \times n_3$, and the initial state must be updated to discard n_2, the control applied at that stage. Hence at each stage the state represents the dimensions of the matrices resulting from the multiplications done so far. The allowable controls at stage k are $u_k \in x_k - \{n_1, n_{N+1}\}$. The system equation evolves according to

$$x_{k+1} = x_k - \{u_k\}.$$

Note that the control will be applied $N-1$ times, therefore the horizon of this problem is $N-1$. The terminal state is $x_{N-1} = \{n_1, n_{N+1}\}$ and the terminal cost is 0. The cost at stage k is given by the number of multiplications,

$$g_k(x_k, u_k) = n_a n_{\overline{u}_k} n_b,$$

where $n_{\overline{u}_k} = u_k$ and

$$a = \max\bigl\{i \in \{1, \ldots, N+1\} \mid i < \overline{u}_k,\ i \in x_k\bigr\},$$

$$b = \min\bigl\{i \in \{1, \ldots, N+1\} \mid i > \overline{u}_k,\ i \in x_k\bigr\}.$$

The DP algorithm for this problem is given by

$$J_{N-1}(x_{N-1}) = 0,$$

$$J_k(x_k) = \min_{u_k \in x_k - \{n_1, n_{N+1}\}} \bigl\{ n_a n_{\overline{u}_k} n_b + J_{k+1}\bigl(x_k - \{u_k\}\bigr) \bigr\}, \qquad k = 0, \ldots, N-2.$$

Now consider the given problem, where $N = 3$ and

$$M_1 \ is\ 2 \times 10,$$

$$M_2 \ is\ 10 \times 5,$$

$$M_3 \ is\ 5 \times 1.$$

The optimal order is $M_1(M_2 M_3)$, requiring 70 multiplications.

(b) In this part we can choose a much simpler state space. Let the state at stage k be given by $\{a, b\}$, where $a, b \in \{1, \ldots, N\}$ and give the indices of the first and the last matrix in the current partial product. There are two possible controls at each stage, which we denote by L and R. Note that L can be applied only when $a \neq 1$ and R can be applied only when $b \neq N$. The system equation evolves according to

$$x_{k+1} = \begin{cases} \{a-1, b\}, & \text{if } u_k = L, \\ \{a, b+1\}, & \text{if } u_k = R, \end{cases} \qquad k = 1, \ldots, N-1.$$

The terminal state is $x_N = \{1, N\}$ with cost 0. The cost at stage k is given by

$$g_k(x_k, u_k) = \begin{cases} n_{a-1} n_a n_{b+1}, & \text{if } u_k = L, \\ n_a n_{b+1} n_{b+2}, & \text{if } u_k = R, \end{cases} \qquad k = 1, \ldots, N-1.$$

For the initial stage, we can take x_0 to be the empty set and $u_0 \in \{1, \ldots, N\}$. The next state will be given by $x_1 = \{u_0, u_0\}$, and the cost incurred at the initial stage will be 0 for all possible controls.

1.19 www

Let $t_1 < t_2 < \cdots < t_{N-1}$ denote the times where $g_1(t) = g_2(t)$. Clearly, it is never optimal to switch functions at any other times. We can therefore divide the problem into $N-1$ stages, where we want to determine for each stage k whether or not to switch activities at time t_k.

Define

$$x_k = \begin{cases} 0 \text{ if on activity } g_1 \text{ just before time } t_k, \\ 1 \text{ if on activity } g_2 \text{ just before time } t_k, \end{cases}$$

$$u_k = \begin{cases} 0 \text{ to continue current activity,} \\ 1 \text{ to switch between activities.} \end{cases}$$

Then the state at time t_{k+1} is simply $x_{k+1} = (x_k + u_k) \bmod 2$, and the profit for stage k is

$$g_k(x_k, u_k) = \int_{t_k}^{t_{k+1}} g_{1+x_{k+1}}(t)dt - u_k c.$$

The DP algorithm is then

$$\begin{aligned} J_N(x_N) &= 0 \\ J_k(x_k) &= \min_{u_k} \left[g_k(x_k, u_k) + J_{k+1}\big((x_k + u_k) \bmod 2\big) \right]. \end{aligned}$$

1.24 www

(a) Consider the problem with the state equal to the number of free rooms. At state $x \geq 1$ with y customers remaining, if the inkeeper quotes a rate r_i, the transition probability is p_i to state $x-1$ (with a reward of r_i) and $1-p_i$ to state x (with a reward of 0). The DP algorithm for this problem starts with the terminal conditions

$$J(x,0) = J(0,y) = 0, \quad \forall\, x \geq 0,\, y \geq 0,$$

and is given by

$$J(x,y) = \max_{i=1,\ldots,m} \big[p_i(r_i + J(x-1,y-1)) + (1-p_i)J(x,y-1)\big], \quad \forall\, x \geq 0.$$

From this equation and the terminal conditions, we can compute sequentially $J(1,1), J(1,2), \ldots, J(1,\overline{y})$ up to any desired integer $\overline{y}$. Then, we can calculate $J(2,1), J(2,2), \ldots, J(2,\overline{y})$, etc.

We first prove by induction on y that for all y, we have

$$J(x,y) \geq J(x-1,y), \quad \forall\, x \geq 1.$$

Indeed this is true for $y = 0$. Assuming this is true for a given y, we will prove that

$$J(x,y+1) \geq J(x-1,y+1), \quad \forall\, x \geq 1.$$

This relation holds for $x = 1$ since $r_i > 0$. For $x \geq 2$, by using the DP recursion, this relation is written as

$$\max_{i=1,\ldots,m} \big[p_i(r_i + J(x-1,y)) + (1-p_i)J(x,y)\big] \geq \max_{i=1,\ldots,m} \big[p_i(r_i + J(x-2,y)) + (1-p_i)J(x-1,y)\big].$$

By the induction hypothesis, each of the terms on the left-hand side is no less than the corresponding term on the right-hand side, so the above relation holds.

The optimal rate is the one that maximizes in the DP algorithm, or equivalently, the one that maximizes

$$p_i r_i + p_i\big(J(x-1,y-1) - J(x,y-1)\big).$$

The highest rate r_m simultaneously maximizes $p_i r_i$ and minimizes p_i. Since

$$J(x-1,y-1) - J(x,y-1) \leq 0,$$

as proved above, we see that the highest rate simultaneously maximizes $p_i r_i$ and $p_i\big(J(x-1,y-1) - J(x,y-1)\big)$, and so it maximizes their sum.

(b) The algorithm given is the algorithm of Exercise 1.22 applied to the problem of part (a). Clearly, it is optimal to accept an offer of r_i if r_i is larger than the threshold

$$\bar{r}(x,y) = J(x,y-1) - J(x-1,y-1).$$

1.25 www

(a) The total net expected profit from the (buy/sell) investment decisions after transaction costs are deducted is

$$E\left\{\sum_{k=0}^{N-1}\big(u_kP_k(x_k)-c\,|u_k|\big)\right\},$$

where

$$u_k=\begin{cases}1 & \text{if a unit of stock is bought at the } k\text{th period,}\\ -1 & \text{if a unit of stock is sold at the } k\text{th period,}\\ 0 & \text{otherwise.}\end{cases}$$

With a policy that maximizes this expression, we simultaneously maximize the expected total worth of the stock held at time N minus the investment costs (including sale revenues).

The DP algorithm is given by

$$J_k(x_k)=\max_{u_k\in\{-1,0,1\}}\Big[u_kP_k(x_k)-c\,|u_k|+E\big\{J_{k+1}(x_{k+1})\mid x_k\big\}\Big],$$

with

$$J_N(x_N)=0,$$

where $J_{k+1}(x_{k+1})$ is the optimal expected profit when the stock price is x_{k+1} at time $k+1$. Since u_k does not influence x_{k+1} and $E\big\{J_{k+1}(x_{k+1})\mid x_k\big\}$, a decision $u_k\in\{-1,0,1\}$ that maximizes $u_kP_k(x_k)-c\,|u_k|$ at time k is optimal. Since $P_k(x_k)$ is monotonically nonincreasing in x_k, it follows that it is optimal to set

$$u_k=\begin{cases}1 & \text{if } x_k\le \underline{x}_k,\\ -1 & \text{if } x_k\ge \overline{x}_k,\\ 0 & \text{otherwise,}\end{cases}$$

where $\underline{x}_k$ and $\overline{x}_k$ are as in the problem statement. Note that the optimal expected profit $J_k(x_k)$ is given by

$$J_k(x_k)=E\left\{\sum_{i=k}^{N-1}\max_{u_i\in\{-1,0,1\}}\big[u_iP_i(x_i)-c\,|u_i|\big]\right\}.$$

(b) Let n_k be the number of units of stock held at time k. If n_k is less that $N-k$ (the number of remaining decisions), then the value n_k should influence the decision at time k. We thus take as state the pair (x_k,n_k), and the corresponding DP algorithm takes the form

$$V_k(x_k,n_k)=\begin{cases}\max_{u_k\in\{-1,0,1\}}\Big[u_kP_k(x_k)-c\,|u_k|+E\big\{V_{k+1}(x_{k+1},n_k+u_k)\mid x_k\big\}\Big] & \text{if } n_k\ge 1,\\ \max_{u_k\in\{0,1\}}\Big[u_kP_k(x_k)-c\,|u_k|+E\big\{V_{k+1}(x_{k+1},n_k+u_k)\mid x_k\big\}\Big] & \text{if } n_k=0,\end{cases}$$

with

$$V_N(x_N,n_N)=0.$$

Note that we have

$$V_k(x_k, n_k) = J_k(x_k), \qquad \text{if } n_k \geq N - k,$$

where $J_k(x_k)$ is given by the formula derived in part (a). Using the above DP algorithm, we can calculate $V_{N-1}(x_{N-1}, n_{N-1})$ for all values of n_{N-1}, then calculate $V_{N-2}(x_{N-2}, n_{N-2})$ for all values of n_{N-2}, etc.

To show the stated property of the optimal policy, we note that $V_k(x_k, n_k)$ is monotonically nondecreasing with n_k, since as n_k decreases, the remaining decisions become more constrained. An optimal policy at time k is to buy if

$$P_k(x_k) - c + E\big\{V_{k+1}(x_{k+1}, n_k + 1) - V_{k+1}(x_{k+1}, n_k) \mid x_k\big\} \geq 0, \tag{1}$$

and to sell if

$$-P_k(x_k) - c + E\big\{V_{k+1}(x_{k+1}, n_k - 1) - V_{k+1}(x_{k+1}, n_k) \mid x_k\big\} \geq 0. \tag{2}$$

The expected value in Eq. (1) is nonnegative, which implies that if $x_k \leq \underline{x}_k$, implying that $P_k(x_k) - c \geq 0$, then the buying decision is optimal. Similarly, the expected value in Eq. (2) is nonpositive, which implies that if $x_k < \overline{x}_k$, implying that $-P_k(x_k) - c < 0$, then the selling decision cannot be optimal. It is possible that buying at a price greater than $\underline{x}_k$ is optimal depending on the size of the expected value term in Eq. (1).

(c) Let m_k be the number of allowed purchase decisions at time k, i.e., m plus the number of sale decisions up to k, minus the number of purchase decisions up to k. If m_k is less than $N-k$ (the number of remaining decisions), then the value m_k should influence the decision at time k. We thus take as state the pair (x_k, m_k), and the corresponding DP algorithm takes the form

$$W_k(x_k, m_k) = \begin{cases} \max_{u_k \in \{-1,0,1\}} \Big[u_k P_k(x_k) - c\,|u_k| + E\big\{W_{k+1}(x_{k+1}, m_k - u_k) \mid x_k\big\}\Big] & \text{if } m_k \geq 1, \\ \max_{u_k \in \{-1,0\}} \Big[u_k P_k(x_k) - c\,|u_k| + E\big\{W_{k+1}(x_{k+1}, m_k - u_k) \mid x_k\big\}\Big] & \text{if } m_k = 0, \end{cases}$$

with

$$W_N(x_N, m_N) = 0.$$

From this point the analysis is similar to the one of part (b).

(d) The DP algorithm takes the form

$$H_k(x_k, m_k, n_k) = \max_{u_k \in \{-1,0,1\}} \Big[u_k P_k(x_k) - c\,|u_k| + E\big\{H_{k+1}(x_{k+1}, m_k - u_k, n_k + u_k) \mid x_k\big\}\Big]$$

if $m_k \geq 1$ and $n_k \geq 1$, and similar formulas apply for the cases where $m_k = 0$ and/or $n_k = 0$ [compare with the DP algorithms of parts (b) and (c)].

(e) Let r be the interest rate, so that x invested dollars at time k will become $(1+r)^{N-k}x$ dollars at time N. Once we redefine the expected profit $P_k(x_k)$ to be

$$P_k(x) = E\{x_N \mid x_k = x\} - (1+r)^{N-k}x,$$

the preceding analysis applies.

1.27 www

We consider part (b), since part (a) is essentially a special case. We will consider the problem of placing $N-2$ points between the endpoints A and B of the given subarc. We will show that the polygon of maximal area is obtained when the $N-2$ points are equally spaced on the subarc between A and B. Based on geometric considerations, we impose the restriction that the angle between any two successive points is no more than π.

As the subarc is traversed in the clockwise direction, we number sequentially the encountered points as $x_1, x_2, \ldots, x_N$, where x_1 and x_N are the two endpoints A and B of the arc, respectively. For any point x on the subarc, we denote by ϕ the angle between x and x_N (measured clockwise), and we denote by $A_k(\phi)$ the maximal area of a polygon with vertices the center of the circle, the points x and x_N, and $N-k-1$ additional points on the subarc that lie between x and x_N.

Without loss of generality, we assume that the radius of the circle is 1, so that the area of the triangle that has as vertices two points on the circle and the center of the circle is $(1/2)\sin u$, where u is the angle corresponding to the center.

By viewing as state the angle ϕ_k between x_k and x_N, and as control the angle u_k between x_k and x_{k+1}, we obtain the following DP algorithm

$$A_k(\phi_k) = \max_{0 \le u_k \le \min\{\phi_k, \pi\}} \left[\frac{1}{2} \sin u_k + A_{k+1}(\phi_k - u_k) \right], \qquad k = 1, \ldots, N-2. \tag{1}$$

Once x_{N-1} is chosen, there is no issue of further choice of a point lying between x_{N-1} and x_N, so we have

$$A_{N-1}(\phi) = \frac{1}{2} \sin \phi, \tag{2}$$

using the formula for the area of the triangle formed by x_{N-1}, x_N, and the center of the circle.

It can be verified by induction that the above algorithm admits the closed form solution

$$A_k(\phi_k) = \frac{1}{2}(N-k) \sin\left(\frac{\phi_k}{N-k}\right), \qquad k = 1, \ldots, N-1, \tag{3}$$

and that the optimal choice for u_k is given by

$$u_k^* = \frac{\phi_k}{N-k}.$$

Indeed, the formula (3) holds for $k = N-1$, by Eq. (2). Assuming that Eq. (3) holds for $k+1$, we have from the DP algorithm (1)

$$A_k(\phi_k) = \max_{0 \le u_k \le \min\{\phi_k, \pi\}} H_k(u_k, \phi_k), \tag{4}$$

where

$$H_k(u_k, \phi_k) = \frac{1}{2}\sin u_k + \frac{1}{2}(N-k-1)\,\sin\left(\frac{\phi_k - u_k}{N-k-1}\right). \tag{5}$$

It can be verified that for a fixed ϕ_k and in the range $0 \le u_k \le \min\{\phi_k, \pi\}$, the function $H_k(\cdot, \phi_k)$ is concave (its second derivative is negative) and its derivative is 0 only at the point $u_k^* = \phi_k/(N-k)$ which must therefore be its unique maximum. Substituting this value of u_k^* in Eqs. (4) and (5), we obtain

$$A_k(\phi_k) = \frac{1}{2}\sin\left(\frac{\phi_k}{N-k}\right) + \frac{1}{2}(N-k-1)\,\sin\left(\frac{\phi_k - \phi_k/(N-k)}{N-k-1}\right) = \frac{1}{2}(N-k)\,\sin\left(\frac{\phi_k}{N-k}\right),$$

and the induction is complete.

Thus, given an optimally placed point x_k on the subarc with corresponding angle ϕ_k, the next point x_{k+1} is obtained by advancing clockwise by $\phi_k/(N-k)$. This process, when started at x_1 with ϕ_1 equal to the angle between x_1 and x_N, yields as the optimal solution an equally spaced placement of the points on the subarc.

Solutions Vol. I, Chapter 2

2.4 www

(a) We denote by P_k the OPEN list *after* having removed k nodes from OPEN, (i.e., after having performed k iterations of the algorithm). We also denote d_j^k the value of d_j at this time. Let $b_k = \min_{j \in P_k}\{d_j^k\}$. First, we show by induction that $b_0 \leq b_1 \leq \cdots \leq b_k$. Indeed, $b_0 = 0$ and $b_1 = \min_j\{a_{s_j}\} \geq 0$, which implies that $b_0 \leq b_1$. Next, we assume that $b_0 \leq \cdots \leq b_k$ for some $k \geq 1$; we shall prove that $b_k \leq b_{k+1}$. Let j_{k+1} be the node removed from OPEN during the $(k+1)$th iteration. By assumption $d_{j_{k+1}}^k = \min_{j \in P_k}\{d_j^k\} = b_k$, and we also have

$$d_i^{k+1} = \min\{d_i^k, d_{j_{k+1}}^k + a_{j_{k+1}i}\}.$$

We have $P_{k+1} = (P_k - \{j_{k+1}\}) \cup N_{k+1}$, where N_{k+1} is the set of nodes i satisfying $d_i^{k+1} = d_{j_{k+1}}^k + a_{j_{k+1}i}$ and $i \notin P_k$. Therefore,

$$\min_{i \in P_{k+1}}\{d_i^{k+1}\} = \min_{i \in (P_k - \{j_{k+1}\}) \cup N_{k+1}}\{d_i^{k+1}\} = \min\left[\min_{i \in P_k - \{j_{k+1}\}}\{d_i^{k+1}\}, \min_{i \in N_{k+1}}\{d_i^{k+1}\}\right].$$

Clearly,

$$\min_{i \in N_{k+1}}\{d_i^{k+1}\} = \min_{i \in N_{k+1}}\{d_{j_{k+1}}^k + a_{j_{k+1}i}\} \geq d_{j_{k+1}}^k.$$

Moreover,

$$\begin{aligned} \min_{i \in P_k - \{j_{k+1}\}}\{d_i^{k+1}\} &= \min_{i \in P_k - \{j_{k+1}\}}\left[\min\{d_i^k, d_{j_{k+1}}^k + a_{j_{k+1}i}\}\right] \\ &\geq \min\left[\min_{i \in P_k - \{j_{k+1}\}}\{d_i^k\}, d_{j_{k+1}}^k\right] = \min_{i \in P_k}\{d_i^k\} = d_{j_{k+1}}^k, \end{aligned}$$

because we remove from OPEN this node with the *minimum* d_i^k. It follows that $b_{k+1} = \min_{i \in P_{k+1}}\{d_i^{k+1}\} \geq d_{j_{k+1}}^k = b_k$.

Now, we may prove that once a node exits OPEN, it never re-enters. Indeed, suppose that some node i exits OPEN after the k^*th iteration of the algorithm; then, $d_i^{k^*-1} = b_{k^*-1}$. If node i re-enters OPEN after the ℓ^*th iteration (with $\ell^* > k^*$), then we have $d_i^{\ell^1-1} > d_i^{\ell^*} = d_{j_\ell^*}^{\ell^*-1} + a_{j_{\ell^*}i} \geq d_{j_{\ell^*}}^{\ell^*-1} = b_{\ell^*-1}$. On the other hand, since d_i is *non-increasing*, we have $b_{k^*-1} = d_i^{k^*-1} \geq d_i^{\ell^*-1}$. Thus, we obtain $b_{k^*-1} > b_{\ell^*-1}$, which contradicts the fact that b_k is non-decreasing.

Next, we claim the following: after the kth iteration, d_i^k equals the length of the shortest possible path from s to node $i \in P_k$ under the restriction that all *intermediate nodes belong to* C_k. The proof will be done by induction on k. For $k = 1$, we have $C_1 = \{s\}$ and $d_i^1 = a_{si}$, and the claim is obviously true. Next, we assume that the claim is true after iterations $1, \ldots, k$; we shall show that it is also true after iteration

$k+1$. The node j_{k+1} removed from OPEN at the $(k+1)$-st iteration satisfies $\min_{i \in P_k}\{d_i^k\} = d^*_{j_{k+1}}$. Notice now that all neighbors of the nodes in C_k belong either to C_k or to P_k.

It follows that the shortest path from s to j_{k+1} either goes through C_k or it exits C_k, then it passes through a node $j^* \in P_k$, and eventually reaches j_{k+1}. In the latter case, the length of this path is at least equal to the length of the shortest path from s to j^* through C_k; by the induction hypothesis, this equals $d^k_{j^*}$, which is at least $d^k_{j_{k+1}}$. It follows that, for node j_{k+1} exiting the OPEN list, $d^k_{j_{k+1}}$ equals the length of the shortest path from s to j_{k+1}. Similarly, all nodes that have exited previously have their current estimate of d_i equal to the corresponding shortest distance from s.

Notice now that

$$d_i^{k+1} = \min\bigl\{d_i^k, d^k_{j_{k+1}} + a_{j_{k+1}i}\bigr\}.$$

For $i \notin P_k$ and $i \in P_{k+1}$ it follows that the only neighbor of i in $C_{k+1} = C_k \cup \{j_{k+1}\}$ is node j_{k+1}; for such a node i, $d_i^k = \infty$, which leads to $d_i^{k+1} = d^k_{j_{k+1}} + a_{j_{k+1}i}$. For $i \neq j_{k+1}$ and $i \in P_k$, the augmentation of C_k by including j_{k+1} offers one more path from s to i through C_{k+1}, namely that through j_{k+1}. Recall that the shortest path from s to i through C_k has length d_i^k (by the induction hypothesis). Thus, $d_i^{k+1} = \min\bigl\{d_1^k, d^k_{j_{k+1}} + a_{j_{k+1}i}\bigr\}$ is the length of the shortest path from s to i through C_{k+1}.

The fact that each node exits OPEN with its current estimate of d_i being equal to its shortest distance from s has been proved in the course of the previous inductive argument.

(b) Since each node enters the OPEN list at most once, the algorithm will terminate in at most $N-1$ iterations. Updating the d_i's during an iteration and selecting the node to exit OPEN requires $O(N)$ arithmetic operations (i.e., a constant number of operations per node). Thus, the total number of operations is $O(N^2)$.

2.6 www

Proposition: If there exists a path from the origin to each node in T, the modified version of the label correcting algorithm terminates with UPPER $< \infty$ and yields a shortest path from the origin to each node in T. Otherwise the algorithm terminates with UPPER $= \infty$.

Proof: The proof is analogous to the proof of Proposition 3.1. To show that this algorithm terminates, we can use the identical argument in the proof of Proposition 3.1.

Now suppose that for some node $t \in T$, there is no path from s to t. Then a node i such that (i,t) is an arc cannot enter the OPEN list because this would establish that there is a path from s to i, and therefore also a path from s to t. Thus, d_t is never changed and UPPER is never reduced from its initial value of ∞.

Suppose now that there is a path from s to each node $t \in T$. Then, since there is a finite number of distinct lengths of paths from s to each $t \in T$ that do not contain any cycles, and each cycle has nonnegative length, there is also a shortest path. For some arbitrary t, let $(s, j_1, j_2, \ldots, j_k, t)$ be a shortest path and let d_t^* be the corresponding shortest distance. We will show that the value of UPPER upon termination must be equal to $d^* = \max_{t \in T} d_t^*$. Indeed, each subpath $(s, j_1, \ldots, j_m), m = 1, \ldots, k$, of the shortest path $(s, j_1, \ldots, j_k, t)$ must be a shortest path from s to j_m. If the value of UPPER is larger than d^* at termination, the same must be true throughout the algorithm, and therefore UPPER will also be larger than the length of all the paths $(s, j_1, \ldots, j_m), m = 1, \ldots, k$, throughout the algorithm, in view of the nonnegative arc length assumption. If, for each $t \in T$, the parent node j_k enters the OPEN list with d_{j_k} equal to the shortest distance from s to j_k, UPPER will be set to d^* in step 2 immediately following the next time the last of the nodes j_k is examined by the algorithm in step 2. It follows that, for some $\bar{t} \in T$, the associated parent node $\bar{j}_k$ will never enter the OPEN list with $d_{\bar{j}_k}$ equal to the shortest distance from s to $\bar{j}_k$. Similarly, and using also the nonnegative length assumption, this means that node $\bar{j}_{k-1}$ will never enter the OPEN list with $d_{\bar{j}_{k-1}}$ equal to the shortest distance from s to $\bar{j}_{k-1}$. Proceeding backwards, we conclude that $\bar{j}_1$ never enters the OPEN list with $d_{\bar{j}_1}$ equal to the shortest distance from s to $\bar{j}_1$ [which is equal to the length of the arc (s, j_1)]. This happens, however, at the first iteration of the algorithm, obtaining a contradiction. It follows that at termination, UPPER will be equal to d^*.

Finally, it can be seen that, upon termination of the algorithm, the path constructed by tracing the parent nodes backward from d to s has length equal to d_t^* for each $t \in T$. Thus the path is a shortest path from s to t.

2.9 www

(a) It was shown in the proof to proposition 3.2 that (P, p) satisfies CS throughout the original algorithm. Note that deleting arcs does not cause CS conditions to no longer hold. Therefore (P, p) satisfies CS throughout this algorithm. It was also shown in the text that if a pair (P, p) satisfies the CS conditions, then the portion of the path P between node s and any node $i \in P$ is a shortest path from s to i. Now, consider any node j that becomes the terminal node of P through an extension using (i, j). If there is a shortest path from s to t that *does not* include node j, then removing the arcs (k, j) where $k \neq i$ yields a graph including the same shortest path. If the only shortest path from s to t *does* include node j, then since P is a shortest path from s to j, there is a shortest path from s to t that has P as its portion to node j. Thus removing the arcs (k, j), where $k \neq i$ yields a graph including a shortest path from the original graph. If node j has no outgoing arcs, then any path from s to t can not include j. Thus removing j yields a graph including the same paths from s to t as in the original graph. Therefore, both types of arc deletions leave the shortest distance from s to t unaffected.

We can view the auction algorithm with graph reduction as follows: an iteration of the original auction algorithm is applied, followed by arc deletions that do not affect the CS conditions and which leaves the shortest distance from s to t unchanged; another iteration of the original auction algorithm is applied to the new graph, followed by arc deletions; and so on. If an iteration of the original auction algorithm yields a path P with t as the terminal node, P is a shortest path from s to t in the latest modified graph. Since we have shown that each new graph has the same shortest distance from s to t as in the original graph, P must also be a shortest path from s to t in the original graph.

Now assume that no iteration of the original auction algorithm ever yields a path P with t as the terminal node; i.e., there is no path from s to t. Assume that the modified algorithm never terminates. Since there are a finite number of arcs and nodes, there can be only a finite number of arc and node deletions. Consider the algorithm after the last deletion. Since there are no more deletions, there must be an outgoing arc (i_1, i_2) from any terminal node i_1 of P. Since the algorithm never terminates, i_2 must eventually be added to P. There must also be an outgoing arc (i_2, i_3) from i_2, and so on. However, there is only a finite number of nodes so some nodes must be repeated, which implies there is a cycle in P. Since there are no arcs incident to s, there must be some arc not part of the cycle that is incident to a node k in the cycle. But then this node has two incoming arcs. When k became the terminal in P, one of these arcs should have been deleted, yielding a contradiction. Thus the algorithm must terminate. Since there is no path from s to t, the algorithm can only terminate by deleting s.

(b) Consider any cycle of zero length. Let j be the first node of the cycle to be a terminal node of path P. Let i be the node preceding j in the path P, and l be the node preceding j in the cycle. All incoming arcs (k, j) of j with $k \neq i$, including arc (l, j), are deleted. Therefore, our problem reduces to one in which there are no cycles of zero length.

(c) The iterations of the modified algorithm applied to the problem of Exercise 2.8 are given below. The first 13 iterations are the same as in the original algorithm, with the exception that at iteration 2, where the path is extended to include node 2, arc $(4, 2)$ is also deleted. As a result of this deletion, after the contraction in iteration 13, the price of node 4 is changed to L, resulting in faster convergence of the algorithm.

Iteration	Path	Price vector p	Action
1	(1)	(0,0,0,0,0)	contraction at 1
2	(1)	(1,0,0,0,0)	extension to 2
3	(1,2)	(1,0,0,0,0)	contraction at 2
4	(1)	(1,1,0,0,0)	contraction at 1
5	(1)	(2,1,0,0,0)	extension to 2
6	(1,2)	(2,1,0,0,0)	extension to 3
7	(1,2,3)	(2,1,0,0,0)	contraction at 3
8	(1,2)	(2,1,1,0,0)	contraction at 2
9	(1)	(2,2,1,0,0)	contraction at 1
10	(1)	(3,2,1,0,0)	extension to 2
11	(1,2)	(3,2,1,0,0)	extension to 3
12	(1,2,3)	(3,2,1,0,0)	extension to 4
13	(1,2,3,4)	(3,2,1,0,0)	contraction at 4
14	(1,2,3)	(3,2,1,L,0)	contraction at 3
15	(1,2)	(3,2,L+1,L,0)	contraction at 2
16	(1)	(3,L+2,L+1,L,0)	contraction at 1
17	(1)	(L+3,L+2,L+1,L,0)	extension to 2
18	(1,2)	(L+3,L+2,L+1,L,0)	extension to 3
19	(1,2,3)	(L+3,L+2,L+1,L,0)	extension to 4
20	(1,2,3,4)	(L+3,L+2,L+1,L,0)	extension to 5
21	(1,2,3,4,5)	(L+3,L+2,L+1,L,0)	done

2.13 www

(a) We first need to show that d_i^k is the length of the shortest k-arc path originating at i, for $i \neq t$. For $k = 1$,

$$d_i^1 = \min_j c_{ij}$$

which is the length of shortest arc out of i. Assume that d_i^{k-1} is the length of the shortest $(k-1)$-arc path out of i. Then

$$d_i^k = \min_j \{c_{ij} + d_j^{k-1}\}$$

If d_i^k is not the length of the shortest k-arc path, the initial arc of the shortest path must pass through a node other than j. This is true since $d_j^{k-1} \leq$ length of any $(k-1)$-step arc out of j. Let ℓ be the alternative node. From the optimality principle

$$\text{distance of path through } \ell \ = c_{i\ell} + d_\ell^{k-1} \leq d_i^k$$

But this contradicts the choice of d_i^k in the DP algorithm. Thus, d_i^k is the length of the shortest k-arc path out of i.

Since $d_t^k = 0$ for all k, once a k-arc path out of i reaches t we have $d_i^\kappa = d_i^k$ for all $\kappa \geq k$. But with all arc lengths positive, d_i^k is just the shortest path from i to t. Clearly, there is some finite k such that the shortest k-path out of i reaches t. If this were not true, the assumption of positive arc lengths implies that the distance from i to t is infinite. Thus, the algorithm will yield the shortest distances in a finite number of steps. We can estimate the number of steps, N_i as

$$N_i \leq \frac{\min_j d_{jt}}{\min_{j,k} d_{jk}}$$

(b) Let $\bar{d}_i^k$ be the distance estimate generated using the initial condition $d_i^0 = \infty$ and $\underline{d}_i^k$ be the estimate generated using the initial condition $d_i^0 = 0$. In addition, let d_i be the shortest distance from i to t.

Lemma:

$$\underline{d}_i^k \leq \underline{d}_i^{k+1} \leq d_i \leq \bar{d}_i^{k+1} \leq \bar{d}_i^k \tag{1}$$

$$\underline{d}_i^k = d_i = \bar{d}_i^k \quad \text{for } k \text{ sufficently large} \tag{2}$$

Proof: Relation (1) follows from the monotonicity property of DP. Note that $\underline{d}_i^1 \geq \underline{d}_i^0$ and that $\bar{d}_i^1 \leq \bar{d}_i^0$. Equation (2) follows immediately from the convergence of DP (given $d_i^0 = \infty$) and from part a).

Proposition: For every k there exists a time T_k such that for all $T \geq T_k$

$$\underline{d}_i^k \leq d_i^T \leq \bar{d}_i^k, \quad i = 1, 2, \ldots, N$$

Proof: The proof follows by induction. For $k = 0$ the proposition is true, given the positive arc length assumption. Assume it is true for a given k. Let $N(i)$ be a set containing all nodes adjacent to i. For every $j \in N(i)$ there exists a time, T_k^j such that

$$\underline{d}_j^k \leq d_j^T \leq \bar{d}_j^k \quad \forall\, T \geq T_k^j$$

Let T' be the first time i updates its distance estimate given that all $d_j^{T_k^j}$, $j \in N(i)$, estimates have arrived. Let d_{ij}^T be the estimate of d_j that i has at time T'. Note that this may differ from $d_j^{T_k^j}$ since the later estimates from j may have arrived before T'. From the Lemma

$$\underline{d}_j^k \leq d_{ij}^{T'} \leq \bar{d}_j^k$$

which, coupled with the monotonicity of DP, implies

$$\underline{d}_i^{k+1} \leq d_i^T \leq \bar{d}_i^{k+1} \quad \forall\, T \geq T'$$

Since each node never stops transmitting, T' is finite and the proposition is proved. Using the Lemma, we see that there is a finite k such that $\underline{d}_i^\kappa = d_i = \bar{d}_i^\kappa$, $\forall\, \kappa \geq k$. Thus, from the proposition, there exists a finite time T^* such that $d_i^T = d_i^*$ for all $T \geq T^*$ and i.

Solutions Vol. I, Chapter 3

3.6 (www)

This problem is similar to the Brachistochrone Problem (Example 4.2) described in the text. As in that problem, we introduce the system

$$\dot{x} = u$$

and have a fixed terminal state problem [$x(0) = a$ and $x(T) = b$]. Letting

$$g(x, u) = \frac{\sqrt{1+u^2}}{Cx},$$

the Hamiltonian is

$$H(x, u, p) = g(x, u) + pu.$$

Minimization of the Hamiltonian with respect to u yields

$$p(t) = -\nabla_u g\big(x(t), u(t)\big).$$

Since the Hamiltonian is constant along an optimal trajectory, we have

$$g\big(x(t), u(t)\big) - \nabla_u g\big(x(t), u(t)\big)\, u(t) = \text{constant}.$$

Substituting in the expression for g, we have

$$\frac{\sqrt{1+u^2}}{Cx} - \frac{u^2}{\sqrt{1+u^2}\, Cx} = \frac{1}{\sqrt{1+u^2}\, Cx} = \text{constant},$$

which simplifies to

$$\big(x(t)\big)^2 \Big(1 + \big(\dot{x}(t)\big)^2\Big) = \text{constant}.$$

Thus an optimal trajectory satisfies the differential equation

$$\dot{x}(t) = \frac{\sqrt{D - \big(x(t)\big)^2}}{\big(x(t)\big)^2}.$$

It can be seen through straightforward calculation that the curve

$$\big(x(t)\big)^2 + (t - d)^2 = D$$

satisfies this differential equation, and thus the curve of minimum travel time from A to B is an arc of a circle.

3.9 www

We have the system $\dot{x}(t) = Ax(t) + Bu(t)$, for which we want to minimize the quadratic cost

$$x(T)'Q_T x(T) + \int_0^T \big(x(t)'Qx(t) + u(t)'Ru(t)\big)dt.$$

The Hamiltonian here is

$$H(x,u,p) = x'Qx + u'Ru + p'(Ax + Bu),$$

and the adjoint equation is

$$\dot{p}(t) = -A'p(t) - 2Qx(t),$$

with the terminal condition

$$p(T) = 2Qx(T).$$

Minimizing the Hamiltonian with respect to u yields the optimal control

$$\begin{aligned} u^*(t) &= \arg\min_u \big[x^*(t)'Qx^*(t) + u'Ru + p'\big(Ax^*(t) + Bu\big)\big] \\ &= \frac{1}{2}R^{-1}B'p(t). \end{aligned}$$

We now hypothesize a linear relation between $x^*(t)$ and $p(t)$

$$2K(t)x^*(t) = p(t), \quad \forall\ t \in [0,T],$$

and show that $K(t)$ can be obtained by solving the Riccati equation. Substituting this value of $p(t)$ into the previous equation, we have

$$u^*(t) = -R^{-1}B'K(t)x^*(t).$$

By combining this result with the system equation, we have

$$\dot{x}(t) = \big(A - BR^{-1}B'K(t)\big)x^*(t). \tag{1}$$

Differentiating $2K(t)x^*(t) = p(t)$ and using the adjoint equation yields

$$2\dot{K}(t)x^*(t) + 2K(t)\dot{x}^*(t) = -A'2K(t)x^*(t) - 2Qx^*(t).$$

Combining with Eq. (1), we have

$$\dot{K}(t)x^*(t) + K(t)\big(A - BR^{-1}B'K(t)\big)x^*(t) = -A'K(t)x^*(t) - Qx^*(t),$$

and we thus see that $K(t)$ should satisfy the Riccati equation

$$\dot{K}(t) = -K(t)A - A'K(t) + K(t)BR^{-1}B'K(t) - Q.$$

From the terminal condition $p(T) = 2Qx(T)$, we have $K(T) = Q$, from which we can solve for $K(t)$ using the Riccati equation. Once we have $K(t)$, we have the optimal control $u^*(t) = -R^{-1}B'K(t)x^*(t)$. By reversing the previous arguments, this control can then be shown to satisfy all the conditions of the Pontryagin Minimum Principle.

Solutions Vol. I, Chapter 4

4.10 www

(a) Clearly, the function J_N is continuous. Assume that J_{k+1} is continuous. We have

$$J_k(x) = \min_{u \in \{0,1,\ldots\}} \big\{cu + L(x+u) + G(x+u)\big\}$$

where

$$G(y) = \underset{w_k}{E}\{J_{k+1}(y - w_k)\}$$
$$L(y) = \underset{w_k}{E}\{p \max(0, w_k - y) + h \max(0, y - w_k)\}$$

Thus, L is continuous. Since J_{k+1} is continuous, G is continuous for bounded w_k. Assume that J_k is not continuous. Then there exists a $\hat{x}$ such that as $y \to \hat{x}$, $J_k(y)$ does not approach $J_k(\hat{x})$. Let

$$u^y = \arg \min_{u \in \{0,1,\ldots\}} \big\{cu + L(y+u) + G(y+u)\big\}$$

Since L and G are continuous, the discontinuity of J_k at $\hat{x}$ implies

$$\lim_{y \to \hat{x}} u^y \neq u^{\hat{x}}$$

But since u^y is optimal for y,

$$\lim_{y \to \hat{x}} \big\{cu^y + L(y + u^y) + G(y + u^y)\big\} < \lim_{y \to \hat{x}} \big\{cu^{\hat{x}} + L(y + u^{\hat{x}}) + G(y + u^{\hat{x}})\big\} = J_k(\hat{x})$$

This contradicts the optimality of $J_k(\hat{x})$ for $\hat{x}$. Thus, J_k is continuous.

(b) Let

$$Y_k(x) = J_k(x+1) - J_k(x)$$

Clearly $Y_N(x)$ is a non-decreasing function. Assume that $Y_{k+1}(x)$ is non-decreasing. Then

$$\begin{aligned}
Y_k(x+\delta) - Y_k(x) = {} & c(u^{x+\delta+1} - u^{x+\delta}) - c(u^{x+1} - u^x) \\
& + L(x+\delta+1+u^{x+\delta+1}) - L(x+\delta+u^{x+\delta}) \\
& - [L(x+1+u^{x+1}) - L(x+u^x)] \\
& + G(x+\delta+1+u^{x+\delta+1}) - G(x+\delta+u^{x+\delta}) \\
& - [G(x+1+u^{x+1}) - G(x+u^x)]
\end{aligned}$$

Since J_k is continuous, $u^{y+\delta} = u^y$ for δ sufficiently small. Thus, with δ small,

$$\begin{aligned} Y_k(x+\delta) - Y_k(x) = {} & L(x+\delta+1+u^{x+1}) - L(x+\delta+u^x) - [L(x+1+u^{x+1}) - L(x+u^x)] \\ & + G(x+\delta+1+u^{x+1}) - G(x+\delta+u^x) - [G(x+1+u^{x+1}) - G(x+u^x)] \end{aligned}$$

Now, since the control and penalty costs are linear, the optimal order given a stock of x is less than the optimal order given $x+1$ stock plus one unit. Thus

$$u^{x+1} \leq u^x \leq u^{x+1} + 1$$

If $u^x = u^{x+1} + 1$, $Y(x+\delta) - Y(x) = 0$ and we have the desired result. Assume that $u^x = u^{x+1}$. Since $L(x)$ is convex, $L(x+1) - L(x)$ is non-decreasing. Using the assumption that $Y_{k+1}(x)$ is non-decreasing, we have

$$\begin{aligned} Y_k(x+\delta) - Y_k(x) = {} & \underbrace{L(x+\delta+1+u^x) - L(x+\delta+u^x) - [L(x+1+u^x) - L(x+u^x)]}_{\geq 0} \\ & \underbrace{\begin{aligned} + \mathop{E}_{w_k}\big\{ & J_{k+1}(x+\delta+1+u^x-w_k) - J_{k+1}(x+\delta+u^x-w_k) \\ & -[J_{k+1}(x+1+u^x-w_k) - J_{k+1}(x+u^x-w_k)]\big\} \end{aligned}}_{\geq 0} \\ \geq 0 & \end{aligned}$$

Thus, $Y_k(x)$ is a non-decreasing function in x.

(c) From their definition and a straightforward induction it can be shown that $J_k^*(x)$ and $J_k(x,u)$ are bounded below. Furthermore,since $\lim_{x\to\infty} L_k(x,u) = \infty$, we obtain $lim_{x\to\infty}(x,0) = \infty$.

From the definition of $J_k(x,u)$, we have

$$J_k(x,u) = J_k(x+1, u-1) + c, \qquad \forall \;\; u \;\in\; \{1,2,\ldots\}. \tag{1}$$

Let S_k be the smallest real number satisfying

$$J_k(S_k, 0) = J_k(S_k+1, 0) + c \tag{2}$$

We show that S_k is well defined. If no S_k satisfying Eq. (2) exists, we must have either $J_k(x,0) - J_k(x+1,0) > c,\;\forall\, x \;\in\; \mathcal{R}$ or $J_k(x,0) - J_k(x+1,0) < 0,\;\forall\, x \;\in\; \mathcal{R}$, because J_k is continuous. The first possibility contradicts the fact that $\lim_{x\to\infty} J_k(x,0) = \infty$. The second possibility implies that $\lim_{x\to\,-\infty} J_k(x,0) + cx$ is finite. However, using the boundedness of $J_{k+1}^*(x)$ from below, we obtain $\lim_{x\to\,-\infty} J_k(x,0) + cx = \infty$. The contradiction shows that S_k is well defined.

We now derive the form of an optimal policy $u_k^*(x)$. Fix some x and consider first the case $x \geq S_k$. Using the fact that $J_k(x,u) - J_k(x+1,u)$ is nondecreasing function of x we have for any $u \;\in\; \{0,1,2,\ldots\}$

$$J_k(x+1,u) - J_k(x,u) \geq J_k(S_k+1,u) J_k(S_k,u) = J_k(S_k+1,0) - J_k(S_k,0) = -c$$

Therefore,

$$J_k(x, u+1) = J_k(x+1, u) + c \geq J_k(x, u) \qquad \forall u \ \in \ \{0, 1, \ldots\}, \ \forall \ x \geq S_k.$$

This shows that $u = 0$ minimizes $J_k(x, u)$, for all $x \geq S_k$. Now let $x \ \in \ [S_k - n, S_k - n + 1), \ n \ \in \ \{1, 2, \ldots\}$. Using Eq. (1), we have

$$J_k(x, n+m) - J_k(x, n) = J_k(x+n, m) - J_k(x+n, 0) \geq 0 \qquad \forall \ m \ \ in \ \{0, 1, \ldots\}. \tag{3}$$

However, if $u < n$ then $x + u < S_k$ and

$$J_k(x+u+1, 0) - J_k(x+u, 0) < J_k(S_k+1, 0) - J_k(S_k, 0) = -c.$$

Therefore,

$$J_k(x, u+1) = J_k(x+u+1, 0) + (u+1)c < J_k(x+u, 0) + uc = J_k(x, u) \qquad \forall \ u \ \in \ \{0, 1, \ldots\}, \ \ n < n. \tag{4}$$

Inequalities (3) and (4) show that $u = n$ minimizes $J_k(x, u)$ whenever $x \ \in \ [S_k - n, S_k - n + 1)$.

4.18 www

Let the state x_k be defined as

$$x_k = \begin{cases} T, & \text{if the selection has already terminated} \\ 1, & \text{if the k}^{\text{th}} \text{ object observed has rank 1} \\ 0, & \text{if the k}^{\text{th}} \text{ object observed has rank} < 1 \end{cases}$$

The system evolves according to

$$x_{k+1} = \begin{cases} T, & \text{if } u_k = \text{stop or } x_k = T \\ w_k, & \text{if } u_k = \text{continue} \end{cases}$$

The cost function is given by

$$g_k(x_k, u_k, w_k) = \begin{cases} \frac{k}{N,} & \text{if } x_k = 1 \text{ and } u_k = \text{stop} \\ 0, & \text{otherwise} \end{cases}$$

$$g_N(x_N) = \begin{cases} 1, & \text{if } x_N = 1 \\ 0, & \text{otherwise} \end{cases}$$

Note that if termination is selected at stage k and $x_k \neq 1$ then the probability of success is 0. Thus, if $x_k = 0$ it is always optimal to continue.

To complete the model we have to determine $P(w_k \mid x_k, u_k) \stackrel{\triangle}{=} P(w_k)$ when the control $u_k =$ continue. At stage k, we have already selected k objects from a sorted set. Since we know nothing else about these objects the new element can, with equal probability, be in any relation with the already observed objects a_j

$$\underbrace{\cdots < a_{i_1} < \cdots < a_{i_2} < \cdots \qquad \cdots < a_{i_k} \cdots}_{k+1 \text{ possible positions for } a_{k+1}}$$

Thus,

$$P(w_k = 1) = \frac{1}{k+1}, \qquad P(w_k = 0) = \frac{k}{k+1}$$

Proposition: If $k \in S_N \triangleq \{i \,|\, \left(\frac{1}{N-1} + \cdots + \frac{1}{i}\right) \leq 1\}$, then

$$J_k(0) = \frac{k}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{k}\right), \qquad J_k(1) = \frac{k}{N}.$$

Proof: For $k = N-1$

$$J_{N-1}(0) = \max\Big[\underbrace{0}_{\text{stop}}, \underbrace{E\{w_{N-1}\}}_{\text{continue}}\Big] = \frac{1}{N}$$

and $\mu^*_{N-1}(0) = \text{continue}$. Also,

$$J_{N-1}(1) = \max\Big[\underbrace{\frac{N-1}{N}}_{\text{stop}}, \underbrace{E\{w_{N-1}\}}_{\text{continue}}\Big] = \frac{N-1}{N}$$

and $\mu^*_{N-1}(1) = \text{stop}$. Note that $N-1 \in S_N$ for all S_N.

Assume the proposition is true for $J_{k+1}(x_{k+1})$. Then

$$J_k(0) = \max\Big[\underbrace{0}_{\text{stop}}, \underbrace{E\{J_{k+1}(w_k)\}}_{\text{continue}}\Big]$$

$$J_k(1) = \max\Big[\underbrace{\frac{k}{N}}_{\text{stop}}, \underbrace{E\{J_{k+1}(w_k)\}}_{\text{continue}}\Big]$$

Now,

$$\begin{aligned} E\{J_{k+1}(w_k)\} &= \frac{1}{k+1}\frac{k+1}{N} + \frac{k}{k+1}\frac{k+1}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{k+1}\right) \\ &= \frac{k}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{k}\right) \end{aligned}$$

Clearly

$$J_k(0) = \frac{k}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{k}\right)$$

and $\mu^*_k(0) = \text{continue}$. If $k \in S_N$,

$$J_k(1) = \frac{k}{N}$$

and $\mu_k^*(1) = \text{stop}$. **Q.E.D.**

Proposition: If $k \notin S_N$

$$J_k(0) = J_k(1) = \frac{\delta - 1}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta - 1}\right)$$

where δ is the minimum element of S_N.

Proof: For $k = \delta - 1$

$$\begin{aligned} J_k(0) &= \frac{1}{\delta}\frac{\delta}{N} + \frac{\delta-1}{\delta}\frac{\delta}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta}\right) \\ &= \frac{\delta-1}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta-1}\right) \end{aligned}$$

$$\begin{aligned} J_k(1) &= \max\left[\frac{\delta-1}{N}, \frac{\delta-1}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta-1}\right)\right] \\ &= \frac{\delta-1}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta-1}\right) \end{aligned}$$

and $\mu_{\delta-1}^*(0) = \mu_{\delta-1}^*(1) = \text{continue}$.

Assume the proposition is true for $J_k(x_k)$. Then

$$J_{k-1}(0) = \frac{1}{k}J_k(1) + \frac{k-1}{k}J_k(0) = J_k(0)$$

and $\mu_{k-1}^*(0) = \text{continue}$.

$$\begin{aligned} J_{k-1}(1) &= \max\left[\frac{1}{k}J_k(1) + \frac{k-1}{k}J_k(0), \frac{k-1}{N}\right] \\ &= \max\left[\frac{\delta-1}{N}\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta-1}\right), \frac{k-1}{N}\right] \\ &= J_k(0) \end{aligned}$$

and $\mu_{k-1}^*(1) = \text{continue}$. **Q.E.D.**

Thus the optimum policy is to continue until the δ^{th} object, where δ is the minimum integer such that $\left(\frac{1}{N-1} + \cdots + \frac{1}{\delta}\right) \leq 1$, and then stop at the first time an element is observed with largest rank.

4.31 (www)

(a) In order that $A_k x + B_k u + w \in X$ for all $w \in W_k$, it is sufficient that $A_k x + B_k u$ belong to some ellipsoid $\tilde{X}$ such that the vector sum of $\tilde{X}$ and W_k is contained in X. The ellipsoid

$$\tilde{X} = \{z \mid z'Fz \leq 1\},$$

where for some scalar $\beta \in (0,1)$,

$$F^{-1} = (1-\beta)(\Psi^{-1} - \beta^{-1} D_k^{-1})$$

has this property (based on the hint and assuming that F^{-1} is well-defined as a positive definite matrix). Thus, it is sufficient that x and u are such that

$$(A_k x + B_k u)' F (A_k x + B_k u) \le 1. \tag{1}$$

In order that for a given x, there exists u with $u' R_k u \le 1$ such that Eq. (1) is satisfied as well as

$$x' \Xi x \le 1,$$

it is sufficient that x is such that

$$\min_{u \in \Re^m} \bigl[x' \Xi x + u' R_k u + (A_k x + B_k u)' F (A_k x + B_k u) \bigr] \le 1, \tag{2}$$

or by carrying out explicitly the quadratic minimization above,

$$x' K x \le 1,$$

where

$$K = A_k' (F^{-1} + B_k R_k^{-1} B_k')^{-1} + \Xi.$$

The control law

$$\mu(x) = -(R_k + B_k' F B_k)^{-1} B_k' F A_k x$$

attains the minimum in Eq. (2) for all x, so it achieves reachability.

(b) Follows by iterative application of the results of part (a), starting with $k = N-1$ and proceeding backwards.

(c) Follows from the arguments of part (a).

Solutions Vol. I, Chapter 5

5.1 www

Define

$$\begin{aligned} y_N &= x_N, \\ y_k &= x_k + A_k^{-1}w_k + A_k^{-1}A_{k+1}^{-1}w_{k+1} + \ldots + A_k^{-1}\cdots A_{N-1}^{-1}w_{N-1}. \end{aligned}$$

Then

$$\begin{aligned} y_k &= x_k + A_k^{-1}(w_k - x_{k+1}) + A_k^{-1}y_{k+1} \\ &= x_k + A_k^{-1}(-A_kx_k - B_ku_k) + A_k^{-1}y_{k+1} \\ &= -A_k^{-1}B_ku_k + A_k^{-1}y_{k+1} \end{aligned}$$

and

$$y_{k+1} = A_ky_k + B_ku_k.$$

Now, the cost function is the expected value of

$$x_N{}'Qx_N + \sum_{k=0}^{N-1} u_k{}'R_ku_k = y_0{}'K_0y_0 + \sum_{k=0}^{N-1}(y_{k+1}{}'K_{k+1}y_{k+1} - y_k{}'K_ky_k + u_k{}'R_ku_k).$$

We have

$$\begin{aligned} y_{k+1}{}'K_{k+1}y_{k+1} - y_k{}'K_ky_k + u_k{}'R_ku_k &= (A_ky_k + B_ku_k)'K_{k+1}(A_ky_k + B_ku_k) + u_k{}'R_ku_k \\ &\quad - y_k{}'A_k{}'[K_{k+1} - K_{k+1}B_k(B_k{}'K_{k+1}B_k)^{-1}B_k\tau rK_{k+1}]A_ky_k \\ &= y_k{}'A_k{}'K_{k+1}A_ky_k + 2y_k'A_k'K_{k+1}B_ku_k + u_k{}'B_k{}'K_{k+1}B_ku_k \\ &\quad - y_k{}'A_k{}'K_{k+1}A_ky_k + y_k{}'A_k{}'K_{k+1}B_kP_k^{-1}B_k'K_{k+1}A_ky_k \\ &\quad + u_k{}'R_ku_k \\ &= -2y_k'L_k'P_ku_k + u_k{}'P_ku_k + y_k{}'L_k{}'P_kL_ky_k \\ &= (u_k - L_ky_k)'P_k(u_k - L_ky_k). \end{aligned}$$

Thus, the cost function can be written as

$$E\left\{y_0{}'K_0y_0 + \sum_{k=0}^{N-1}(u_k - L_ky_k)'P_k(u_k - L_ky_k)\right\}.$$

The problem now is to find $\mu_k^*(I_k)$, $k = 0, 1, \ldots, N-1$, that minimize over admissible control laws $\mu_k(I_k)$, $k = 0, 1, \ldots, N-1$, the cost function

$$E\left\{ {y_0}' K_0 y_0 + \sum_{k=0}^{N-1} \big(\mu_k(I_k) - L_k y_k\big)' P_k \big(\mu_k(I_k) - L_k y_k\big) \right\}.$$

We do this minimization by first minimizing over μ_{N-1}, then over μ_{N-2}, etc. The minimization over μ_{N-1} involves just the last term in the sum and can be written as

$$\begin{aligned} &\min_{\mu_{N-1}} E\Big\{ \big(\mu_{N-1}(I_{N-1}) - L_{N-1} y_{N-1}\big)' P_{N-1} \big(\mu_{N-1}(I_{N-1}) - L_{N-1} y_{N-1}\big) \Big\} \\ &\qquad = E\Big\{ \min_{u_{N-1}} E\Big\{ \big(u_{N-1} - L_{N-1} y_{N-1}\big)' P_{N-1} \big(u_{N-1} - L_{N-1} y_{N-1}\big) \big| I_{N-1} \Big\} \Big\}. \end{aligned}$$

Thus this minimization yields the optimal control law for the last stage:

$$\mu_{N-1}^*(I_{N-1}) = L_{N-1}\, E\Big\{ y_{N-1} \big| I_{N-1} \Big\}.$$

[Recall here that, generically, $E\{z|I\}$ minimizes over u the expression $E_z\big\{(u-z)'P(u-z)|\ |\big\}$ for any random variable z, any conditioning variable I, and any positive semidefinite matrix P.] The minimization over μ_{N-2} involves

$$\begin{aligned} &E\Big\{ \big(\mu_{N-2}(I_{N-2}) - L_{N-2} y_{N-2}\big)' P_{N-2} \big(\mu_{N-2}(I_{N-2}) - L_{N-2} y_{N-2}\big) \Big\} \\ &\qquad + E\Big\{ \big(E\{y_{N-1}|I_{N-1}\} - y_{N-1}\big)' L_{N-1}' P_{N-1} L_{N-1} \big(E\{y_{N-1}|I_{N-1}\} - y_{N-1}\big) \Big\}. \end{aligned}$$

However, as in Lemma 5.2.1, the term $E\{y_{N-1}|I_{N-1}\} - y_{N-1}$ does not depend on any of the controls (it is a function of $x_0, w_0, \ldots, w_{N-2}, v_0, \ldots, v_{N-1}$). Thus the minimization over μ_{N-2} involves just the first term above and yields similarly as before

$$\mu_{N-2}^*(I_{N-2}) = L_{N-2}\, E\Big\{ y_{N-2} \big| I_{N-2} \Big\}.$$

Proceeding similarly, we prove that for all k

$$\mu_k^*(I_k) = L_k\, E\Big\{ y_k \big| I_k \Big\}.$$

Note: The preceding proof can be used to provide a quick proof of the separation theorem for linear-quadratic problems in the case where $x_0, w_0, \ldots, w_{N-1}, v_0, \ldots, v_{N-1}$ are independent. If the cost function is

$$E\left\{ {x_N}' Q_N x_N + \sum_{k=0}^{N-1} \big({x_k}' Q_k x_k + {u_k}' R_k u_k \big) \right\}$$

the preceding calculation can be modified to show that the cost function can be written as

$$E\left\{ {x_0}' K_0 x_0 + \sum_{k=0}^{N-1} \big((u_k - L_k x_k)' P_k (u_k - L_k x_k) + {w_k}' K_{k+1} w_k \big) \right\}.$$

By repeating the preceding proof we then obtain the optimal control law as

$$\mu_k^*(I_k) = L_k\, E\big\{ x_k \big| I_k \big\}$$

5.3 www

The control at time k is (u_k, α_k), where α_k is a variable taking values 1 (if the next measurement at time $k+1$ is of type 1) or 2 (if the next measurement is of type 2). The cost functional is

$$E\left\{ x_N{}'Qx_N + \sum_{k=0}^{N-1}(x_k{}'Qx_k + u_k{}'Ru_k) + \sum_{k=0}^{N-1} g_{\alpha_k} \right\}.$$

We apply the DP algorithm for $N = 2$. We have from the Riccati equation

$$\begin{aligned} J_1(I_1) &= J_1(z_0, z_1, u_0, \alpha_0) \\ &= \mathop{E}_{x_1}\bigl\{x_1{}'(A'QA + Q)x_1 \,|\, I_1\bigr\} + \mathop{E}_{w_1}\{w'Qw\} \\ &\quad + \min_{u_1}\bigl\{u_1{}'(B'QB + R)u_1 + 2\,E\{x_1 \,|\, I_1\}'A'QBu_1\bigr\} \\ &\quad + \min[g_1, g_2]. \end{aligned}$$

So

$$\mu_1^*(I_1) = -(B'QB + R)^{-1}B'QA\,E\{x_1 \,|\, I_1\},$$

$$\alpha_1^*(I_1) = \begin{cases} 1, & \text{if } g_1 \le g_2, \\ 2, & \text{otherwise.} \end{cases}$$

Note that the measurement selected at $k = 1$ does not depend on I_1. This is intuitively clear since the measurement z_2 will not be used by the controller, so its selection should be based on measurement cost alone and not on the basis of the quality of estimate. The situation is different once more than one stage is considered.

Using a simple modification of the analysis in Section 5.2 of the text, we have

$$\begin{aligned} J_0(I_0) &= J_0(z_0) \\ &= \min_{u_0}\Bigl\{ \mathop{E}_{x_0,w_0}\bigl\{x_0{}'Qx_0 + u_0{}'Ru_0 + Ax_0 + Bu_0 + w_0{}'K_0Ax_0 + Bu_0 + w_0 \,\big|\, z_0\bigr\}\Bigr\} \\ &\quad + \min_{\alpha_0}\Bigl[\mathop{E}_{z_1}\Bigl\{\mathop{E}_{x_1}\bigl\{[x_1 - E\{x_1 \,|\, I_1\}]'P_1[x_1 - E\{x_1 \,|\, I_1\}] \,\big|\, I_1\bigr\} \,\Big|\, z_0, u_0, \alpha_0\Bigr\} + g_{\alpha_0}\Bigr] \\ &\quad + \mathop{E}_{w_1}\{w_1{}'Qw_1\} + \min[g_1, g_2]. \end{aligned}$$

Note that the minimization of the second bracket is indicated only with respect to α_0 and not u_0. The reason is that quantity in the second bracket is the error covariance of the estimation error (weighted by P_1) and, as shown in the text, it does not depend on u_0. Because all stochastic variables are Gaussian, the quantity in the second bracket does not depend on z_0. (The weighted error covariance produced by the Kalman filter is precomputable and depends only on the system and measurement matrices and noise covariances but not on the measurements received.) In fact

$$\begin{aligned} &\mathop{E}_{z_1}\Bigl\{\mathop{E}_{x_1}\bigl\{[x_1 - E\{x_1 \,|\, I_1\}]'P_1[x_1 - E\{x_1 \,|\, I_1\}] \,\big|\, I_1\bigr\} \,\Big|\, z_0, u_0, \alpha_0\Bigr\} \\ &\qquad = \begin{cases} Tr\left(P_1^{\frac{1}{2}} \sum_{1|1}^{1} P_1^{\frac{1}{2}}\right), & \text{if } \alpha_0 = 1, \\ Tr\left(P_1^{\frac{1}{2}} \sum_{1|1}^{2} P_1^{\frac{1}{2}}\right), & \text{if } \alpha_0 = 2, \end{cases} \end{aligned}$$

where $Tr(\cdot)$ denotes the trace of a matrix, and $\sum_{1|1}^{1}$ ($\sum_{1|1}^{2}$) denotes the error covariance of the Kalman filter estimate if a measurement of type 1 (type 2) is taken at $k = 0$. Thus at time $k = 0$, we have that the optimal measurement chosen does not depend on z_0 and is of type 1 if

$$Tr\left(P_1^{\frac{1}{2}}\Sigma_{1|1}^{1}P_1^{\frac{1}{2}}\right) + g_1 \leq Tr\left(P_1^{\frac{1}{2}}\Sigma_{1|1}^{2}P_1^{\frac{1}{2}}\right) + g_2$$

and is of type 2 otherwise.

5.7 www

Exercise 5.7

a) We have

$$\begin{aligned}
p_{k+1}^{j} &= P(x_{k+1} = j \mid z_0, \ldots, z_{k+1}, u_0, \ldots, u_k) \\
&= P(x_{k+1} = j \mid I_{k+1}) \\
&= \frac{P(x_{k+1} = j, z_{k+1} \mid I_k, u_k)}{P(z_{k+1} \mid I_k, u_k)} \\
&= \frac{\sum_{i=1}^{n} P(x_k = i)P(x_{k+1} = j \mid x_k = i, u_k)P(z_{k+1} \mid u_k, x_{k+1} = j)}{\sum_{s=1}^{n}\sum_{i=1}^{n} P(x_k = i)P(x_{k+1} = s \mid x_k = i, u_k)P(z_{k+1} \mid u_k, x_{k+1} = s)} \\
&= \frac{\sum_{i=1}^{n} p_k^i p_{ij}(u_k) r_j(u_k, z_{k+1})}{\sum_{s=1}^{n}\sum_{i=1}^{n} p_k^i p_{is}(u_k) r_s(u_k, z_{k+1})}.
\end{aligned}$$

Rewriting p_{k+1}^{j} in vector form, we have

$$p_{k+1}^{j} = \frac{r_j(u_k, z_{k+1})[P(u_k)'P_k]_j}{\sum_{s=1}^{n} r_s(u_k, z_{k+1})[P(u_k)'P_k]_s}, \qquad j = 1, \ldots, n.$$

Therefore,

$$P_{k+1} = \frac{[r(u_k, z_{k+1})] * [P(u_k)'P_k]}{r(u_k, z_{k+1})'P(u_k)'P_k}.$$

b) The DP algorithm for this system is:

$$\begin{aligned}
\bar{J}_{N-1}(P_{N-1}) &= \min_u \left\{ \sum_{i=1}^{n} p_{N-1}^{i} \sum_{j=1}^{n} p_{ij}(u) g_{N-1}(i, u, j) \right\} \\
&= \min_u \left\{ \sum_{i=1}^{n} p_{N-1}^{i} \big[G_{N-1}(u)\big]_i \right\} \\
&= \min_u \big\{ P_{N-1}' G_{N-1}(u) \big\}
\end{aligned}$$

$$
\begin{aligned}
\bar{J}_k(P_k) &= \min_u \left\{ \sum_{i=1}^{n} p_k^i \sum_{j=1}^{n} p_{ij}(u) g_k(i,u,j) + \sum_{i=1}^{n} p_k^i \sum_{j=1}^{n} p_{ij}(u) \sum_{\theta=1}^{q} r_j(u,\theta) \bar{J}_{k+1}(P_{k+1} \mid P_k, u, \theta) \right\} \\
&= \min_u \left\{ P_k' G_k(u) + \sum_{\theta=1}^{q} r(u,\theta)' P(u)' P_k \bar{J}_{k+1} \left[\frac{[r(u,\theta)] * [P(u)' P_k]}{r(u,\theta)' P(u)' P_k} \right] \right\}.
\end{aligned}
$$

c) For $k = N-1$,

$$
\begin{aligned}
\bar{J}_{N-1}(\lambda P_{N-1}') &= \min_u \big\{ \lambda P_{N-1}' G_{N-1}(u) \big\} \\
&= \min_u \{ \sum_{i=1}^{n} \lambda p_{N-1}^i \big[G_{N-1}(u) \big]_i \} \\
&= \min_u \{ \lambda \sum_{i=1}^{n} p_{N-1}^i [G_{N-1}(u)]_i \} \\
&= \lambda \min_u \{ \sum_{i=1}^{n} p_{N-1}^i [G_{N-1}(u)]_i \} \\
&= \lambda \min_u \{ \sum_{i=1}^{n} p_{N-1}^i [G_{N-1}(u)]_i \} \\
&= \lambda \bar{J}_{N-1}(P_{N-1}).
\end{aligned}
$$

Now assume $\bar{J}_k(\lambda P_k) = \lambda \bar{J}_k(P_k)$. Then,

$$
\begin{aligned}
\bar{J}_{k-1}(\lambda P_{k-1}') &= \min_u \left\{ \lambda P_{k-1}' G_{k-1}(u) + \sum_{\theta=1}^{q} r(u,\theta)' P(u)' \lambda P_{k-1} \bar{J}_k(P_k | P_{k-1}, u, \theta) \right\} \\
&= \min_u \left\{ \lambda P_{k-1}' G_{k-1}(u) + \lambda \sum_{\theta=1}^{q} r(u,\theta)' P(u)' P_{k-1} \bar{J}_k(P_k | P_{k-1}, u, \theta) \right\} \\
&= \lambda \min_u \left\{ P_{k-1}' G_{k-1}(u) + \sum_{\theta=1}^{q} r(u,\theta)' P(u)' P_{k-1} \bar{J}_k(P_k | P_{k-1}, u, \theta) \right\} \\
&= \lambda \bar{J}_{k-1}(P_{k-1}). \qquad \text{Q.E.D.}
\end{aligned}
$$

For any u, $r(u,\theta)' P(u)' P_k$ is a scalar. Therefore, letting $\lambda = r(u,\theta)' P(u)' P_k$, we have

$$
\begin{aligned}
\bar{J}_k(P_k) &= \min_u \left\{ P_k' G_k(u) + \sum_{\theta=1}^{q} r(u,\theta)' P(u)' P_k \bar{J}_{k+1} \left[\frac{[r(u,\theta)] * [P(u)' P_k]}{r(u,\theta)' P(u)' P_k} \right] \right\} \\
&= \min_u \left[P_k' G_k(u) + \sum_{\theta=1}^{q} \bar{J}_{k+1}([r(u,\theta)] * [P(u)' P_k]) \right].
\end{aligned}
$$

d) For $k = N-1$, we have $\bar{J}_{N-1}(P_{N-1}) = \min_u [P_{N-1}' G_{N-1}(u)]$, and so $\bar{J}_{N-1}(P_{N-1})$ has the desired form

$$
\bar{J}_{N-1}(P_{N-1}) = \min \big[P_{N-1}' \alpha_{N-1}^1, \ldots, P_{N-1}' \alpha_{N-1}^m \big],
$$

where $\alpha_{N-1}^j = G_{N-1}(u^j)$ and u^j is the jth element of the control constraint set.

Assume that

$$\bar{J}_{k+1}(P_{k+1}) = \min\big[P'_{k+1}\alpha^1_{k+1}, \ldots, P'_{k+1}\alpha^{m_{k+1}}_{k+1}\big].$$

Then, using the expression from part (c) for $\bar{J}_k(P_k)$,

$$\begin{aligned}
\bar{J}_k(P_k) &= \min_u \left[P'_k G_k(u) + \sum_{\theta=1}^{q} \bar{J}_{k+1}\big([r(u,\theta)] * [P(u)'P_k]\big)\right] \\
&= \min_u \left[P'_k G_k(u) + \sum_{\theta=1}^{q} \min_{m=1,\ldots,m_{k+1}} \left[\big\{[r(u,\theta)] * [P(u)'P_k]\big\}'\alpha^m_{k+1}\right]\right] \\
&= \min_u \left[P'_k G_k(u) + \sum_{\theta=1}^{q} \min_{m=1,\ldots,m_{k+1}} \big[P'_k P(u) r(u,\theta)'\alpha^m_{k+1}\big]\right] \\
&= \min_u \left[P'_k \left\{G_k(u) + \sum_{\theta=1}^{q} \min_{m=1,\ldots,m_{k+1}} \left[P(u) r(u,\theta)'\alpha^m_{k+1}\right]\right\}\right] \\
&= \min\big[P'_k\alpha^1_k, \ldots, P'_k\alpha^{m_k}_k\big],
\end{aligned}$$

where $\alpha^1_k, \ldots, \alpha^{m_k}_k$ are all possible vectors of the form

$$G_k(u) + \sum_{\theta=1}^{q} P(u) r(u,\theta)'\alpha^m_{k+1},$$

as u ranges over the finite set of controls $\{u^1, \ldots, u^m\}$, θ ranges over the set of observation vector indexes $\{1, \ldots, q\}$, and m ranges over the set of indexes $\{1, \ldots, m_{k+1}\}$. The induction is thus complete.

For a quick way to understand the preceding proof, based on polyhedral concavity notions, note that the conclusion is equivalent to asserting that $\bar{J}_k(P_k)$ is a positively homogeneous, concave polyhedral function. The preceding induction argument amounts to showing that the DP formula of part (c) preserves the positively homogeneous, concave polyhedral property of $\bar{J}_{k+1}(P_{k+1})$. This is indeed evident from the formula, since taking minima and nonnegative weighted sums of positively homogeneous, concave polyhedral functions results in a positively homogeneous, concave polyhedral function.

Solutions Vol. I, Chapter 6

6.8 www

First, we notice that $\alpha - \beta$ pruning is applicable only for arcs that point to *right* children, so that at least one sequence of moves (starting from the current position and ending at a terminal position, that is, one with no children) has been considered. Furthermore, due to depth-first search the score at the ancestor positions has been derived without taking into account the positions that can be reached from the current point. Suppose now that α-pruning applies at a position with Black to play. Then, if the current position is reached (due to a move by White), Black can respond in such a way that the final position will be worse (for White) than it would have been if the current position were not reached. What α-pruning saves is searching for even worse positions (emanating from the current position). The reason for this is that White will never play so that Black reaches the current position, because he certainly has a better alternative. A similar argument applies for β pruning.

A second approach: Let us suppose that it is the WHITE's turn to move. We shall prove that a $\beta-$cutoff occurring at the nth position will not affect the backed up score. We have from the definition of β $\beta = \min\{TBS$ of all ancestors of n (white) where BLACK has the move$\}$. For a cutoff to occur: $TBS(n) > \beta$. Observe first of all that $\beta = TBS(n_1)$ for some ancestor n_1 where BLACK has the move. Then there exists a path $n_1, n_2, \ldots, n_k, n$. Since it is WHITE's move at n we have that $TBS(n) = \max\{TBS(n), BS(n_i)\} > \beta$, where n_i are the descendants of n. Consider now a position n_k. Then $TBS(n_k)$ will either remain unchanged or will increase to a value greater than β as a result of the exploration of node n. Proceeding similarly, we conclude that $TBS(n_2)$ will either remain the same or change to a value greater than β. Finally at node n_1 we have that $TBS(n_1)$ will not change since it is BLACK's turn to move and he will choose the move with minimum score. Thus the backed up score and the choice of the next move are unaffected from $\beta-$pruning. A similar argument holds for $\alpha-$pruning.

6.12 www

(a) We have for all i

$$F(i) \geq \hat{F}(i) = a_{ij(i)} + F\big(j(i)\big). \tag{1}$$

Assume, in order to come to a contradiction, that the graph of the $N-1$ arcs $\big(i, j(i)\big)$, $i = 1, \ldots, N-1$, contains a cycle $(i_1, i_2, \ldots, i_k, i_1)$. Using Eq. (1), we have

$$F(i_1) \geq a_{i_1 i_2} + F(i_2),$$

$$F(i_2) \geq a_{i_2 i_3} + F(i_3),$$

$$\cdots$$

$$F(i_k) \geq a_{i_k i_1} + F(i_1).$$

By adding the above inequalities, we obtain

$$0 \geq a_{i_1 i_2} + a_{i_2 i_3} + \cdots + a_{i_k i_1}.$$

Thus the length of the cycle $(i_1, i_2, \ldots, i_k, i_1)$ is nonpositive, a contradiction. Hence, the graph of the $N-1$ arcs $\big(i, j(i)\big)$, $i = 1, \ldots, N-1$, contains no cycle. Given any node $i \neq N$, we can start with arc $\big(i, j(i)\big)$, append the outgoing arc from $j(i)$, and continue up to reaching N (if we did not reach N, a cycle would be formed). The corresponding path, called P_i, is unique since there is only one arc outgoing from each node.

Let $P_i = (i, i_1, i_2, \ldots, i_k, N)$ [so that $i_1 = j(i_1), i_2 = j(i_1), \ldots, N = j(i_k)$]. We have using the hypothesis $\hat{F}(i) \leq F(i)$ for all i

$$\hat{F}(i) = a_{ii_1} + F(i_1) \geq a_{ii_1} + \hat{F}(i_1),$$

and similarly

$$\hat{F}(i_1) = a_{i_1 i_2} + F(i_2) \geq a_{i_1 i_2} + \hat{F}(i_2),$$

$$\cdots$$

$$\hat{F}(i_k) = a_{i_k N} + F(i_N) = a_{i_k N}.$$

By adding the above relations, we obtain

$$\hat{F}(i) \geq a_{ii_1} + a_{i_1 i_2} + \cdots + a_{i_k N}.$$

The result follows since the right-hand side is the length of P_i.

(b) For a counterexample to part (a) in the case where there are cycles of zero length, take $a_{ij} = 0$ for all (i, j), let $F(i) = 0$ for all i, let $(i_1, i_2, \ldots, i_k, i_1)$ be a cycle, and choose $j(i_1) = i_2, \ldots, j(i_{k-1}) = i_k, j(i_k) = i_1$.

(c) We have

$$\hat{F}(i) = \min_{j \in J_i} \big[a_{ij} + F(j)\big] \leq a_{ij_i} + F(j_i) \leq F(i).$$

(d) Consider the heuristic which at node i, generates the path $\overline{P}_i$. Then P_i is the path generated by the rollout algorithm based on this heuristic. The inequality $F(i) \geq a_{ij_i} + F(j_i)$ implies that the rollout algorithm is sequentially improving, so the rollout algorithm yields no worse cost than the base heuristic starting from any node, i.e., the length of P_i is less or equal to the length of $\overline{P}_i$.

(e) Using induction, we can show that after each iteration of the label correcting method, we have for all $i = 1, \ldots, N-1$,

$$F(i) \geq \min_{\{j \mid (i,j) \text{ is an arc}\}} \big[a_{ij} + F(j)\big],$$

and if $F(i) < \infty$, then $F(i)$ is equal to the length of some path starting at i and ending at N. Furthermore, for the first arc (i, j_i) of this path, we have

$$F(i) \geq a_{ij_i} + F(j_i).$$

Thus the assumptions of part (c) are satisfied.

Solutions Vol. I, Chapter 7

7.8 www

A threshold policy is specified by a threshold integer m and has the form

Process the orders if and only if their number exceeds m.

The cost function corresponding to a threshold policy specified by m will be denoted by J_m. By Prop. 3.1(c), this cost function is the unique solution of system of equations

$$J_m(i) = \begin{cases} K + \alpha(1-p)J_m(0) + \alpha p J_m(1) & \text{if } i > m, \\ ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1) & \text{if } i \le m. \end{cases} \tag{1}$$

Thus for all $i \le m$, we have

$$J_m(i) = \frac{ci + \alpha p J_m(i+1)}{1 - \alpha(1-p)},$$

$$J_m(i-1) = \frac{c(i-1) + \alpha p J_m(i)}{1 - \alpha(1-p)}.$$

From these two equations it follows that for all $i \le m$, we have

$$J_m(i) \le J_m(i+1) \quad \Rightarrow \quad J_m(i-1) < J_m(i). \tag{2}$$

Denote now

$$\gamma = K + \alpha(1-p)J_m(0) + \alpha p J_m(1).$$

Consider the policy iteration algorithm, and a policy $\overline{\mu}$ that is the successor policy to the threshold policy corresponding to m. This policy has the form

Process the orders if and only if

$$K + \alpha(1-p)J_m(0) + \alpha p J_m(1) \le ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1)$$

or equivalently

$$\gamma \le ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1).$$

In order for this policy to be a threshold policy, we must have for all i

$$\gamma \le c(i-1) + \alpha(1-p)J_m(i-1) + \alpha p J_m(i) \quad \Rightarrow \quad \gamma \le ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1). \tag{3}$$

This relation holds if the function J_m is monotonically nondecreasing, which from Eqs. (1) and (2) will be true if $J_m(m) \le J_m(m+1) = \gamma$.

Let us assume that the opposite case holds, where $\gamma < J_m(m)$. For $i > m$, we have $J_m(i) = \gamma$, so that

$$ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1) = ci + \alpha\gamma. \tag{4}$$

We also have

$$J_m(m) = \frac{cm + \alpha p \gamma}{1 - \alpha(1-p)},$$

from which, together with the hypothesis $J_m(m) > \gamma$, we obtain

$$cm + \alpha\gamma > \gamma. \tag{5}$$

Thus, from Eqs. (4) and (5) we have

$$ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1) > \gamma, \qquad \text{for all } i > m, \tag{6}$$

so that Eq. (3) is satisfied for all $i > m$.

For $i \leq m$, we have $ci + \alpha(1-p)J_m(i) + \alpha p J_m(i+1) = J_m(i)$, so that the desired relation (3) takes the form

$$\gamma \leq J_m(i-1) \quad \Rightarrow \quad \gamma \leq J_m(i). \tag{7}$$

To show that this relation holds for all $i \leq m$, we argue by contradiction. Suppose that for some $i \leq m$ we have $J_m(i) < \gamma \leq J_m(i-1)$. Then since $J_m(m) > \gamma$, there must exist some $\bar{i} > i$ such that $J_m(\bar{i} - 1) < J_m(\bar{i})$. But then Eq. (2) would imply that $J_m(j-1) < J_m(j)$ for all $j \leq \bar{i}$, contradicting the relation $J_m(i) < \gamma \leq J_m(i-1)$ assumed earlier. Thus, Eq. (7) holds for all $i \leq m$ so that Eq. (3) holds for all i. The proof is complete.

7.12 www

Let Assumption 2.1 hold and let $\pi = \{\mu_0, \mu_1, \ldots\}$ be an admissible policy. Consider also the sets $S_k(i)$ given in the hint with $S_0(i) = \{i\}$. If $t \in S_n(i)$ for all π and i, we are done. Otherwise, we must have for some π and i, and some $k < n$, $S_k(i) = S_{k+1}(i)$ while $t \notin S_k(i)$. For $j \in S_k(i)$, let $m(j)$ be the smallest integer m such that $j \in S_m$. Consider a stationary policy μ with $\mu(j) = \mu_{m(j)}(j)$ for all $j \in S_k(i)$. For this policy we have for all $j \in S_k(i)$,

$$p_{jl}(\mu(j)) > 0 \quad \Rightarrow \quad l \in S_k(i).$$

This implies that the termination state t is not reachable from all states in $S_k(i)$ under the stationary policy μ, and contradicts Assumption 2.1.