# *Dynamic Programming and Optimal Control*

VOL. I, FOURTH EDITION

Dimitri P. Bertsekas

**Massachusetts Institute of Technology**

# *Selected Theoretical Problem Solutions*

Last Updated 2/11/2017

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**

# *Solutions Vol. I, Chapter 1*

**1.8 (Ordering Matrix Multiplications)** www

Each state is a set $S_k \subseteq \{2, \ldots N\}$. The allowable states at stage $k$ are those of cardinality $k$. The allowable controls are $u_k \in \{2, \ldots, N\} - S_k$. This control represents the multiplication of the term ending in $M_{u_k - 1}$ by the one starting in $M_{u_k}$. The system equation evolves according to

$$S_{k+1} = S_k \cup u_k$$

The terminal state is $S_N = \{2, \ldots, N\}$, with cost 0. The cost at stage $k$ is given by the number of multiplications

$$g_k(S_k, u_k) = n_a n_{u_k} n_b$$

where

$$a = \max\bigl\{i \in \{1, \ldots, N+1\} \mid i \notin S_k, i < u_k\bigr\}$$
$$b = \min\bigl\{i \in \{1, \ldots, N+1\} \mid i \notin S_k, i > u_k\bigr\}$$

For example, let $N = 3$ and

$$M_1 \ be \ 1 \times 10$$
$$M_2 \ be \ 10 \times 1$$
$$M_3 \ be \ 1 \times 10$$

The order $(M_1 M_2) M_3$ corresponds to controls $u_1 = 2$ and $u_2 = 3$, giving cost

$$\begin{array}{lll} (u_1 = 2) & n_1 n_2 n_3 = 10 & (a = 1, b = 3) \\ (u_2 = 3) & n_1 n_3 n_4 = 10 & (a = 1, b = 4) \end{array}$$

with a total cost of 20

whereas $M_1(M_2 M_3)$ gives

$$\begin{array}{lll} (u_1 = 3) & n_2 n_3 n_4 = 100 & (a = 2, b = 4) \\ (u_2 = 2) & n_1 n_2 n_4 = 100 & (a = 1, b = 4) \end{array}$$

with a total cost of 200

Now consider the given problem

$$M_1 \ is \ 2 \times 10$$
$$M_2 \ is \ 10 \times 5$$
$$M_3 \ is \ 5 \times 1$$

The shortest path is $\emptyset \rightarrow \{3\} \rightarrow \{2, 3\}$, corresponding to $M_1(M_2 M_3)$ and 70 multiplications.

**1.16** (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 cost

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

and is given by

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

$$J(0,y) = 0.$$

We now prove the last assertion. We first prove by induction on $y$ that for all $y$,

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

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

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

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, and so the above relation holds.

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

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

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

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

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

(b) Clearly, it is optimal to accept an offer of $r_i$ if $r_i$ is larger than the threshold

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

**1.17 (Investing in a Stock)** (www)

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

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

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

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

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 $u_k P_k(x_k) - c\,|u_k|$ 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

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

(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

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

with

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

Note that we have

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

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.

**1.18 (Regular Polygon Theorem)** 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 $\pi$.

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 $\phi$ 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 $\phi_k$ 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 $\phi_k$ and in the range $0 \leq u_k \leq \min\{\phi_k, \pi\}$, 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 $\phi_k$, the next point $x_{k+1}$ is obtained by advancing clockwise by $\phi_k/(N-k)$. This process, when started at $x_1$ with $\phi_1$ 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.

**1.20** (www)

Let $t_1 < t_2 < \ldots < t_{N-1}$ 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

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

**1.27 (Semilinear Systems)** (www)

The DP algorithm is

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

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

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

$$\begin{aligned}
J_k(x_k) &= \min_{u_k} \mathop{E}_{w_k, A_k} \left\{c'_k x_k + g_k(u_k) + b'_{k+1}A_k x_k + b'_{k+1}f_k(u_k) + b'_{k+1}w_k + d_{k+1}\right\} \\
&= c'_k x_k + b'_{k+1}\, E\{A_k\}\, x_k + b'_{k+1}\, E\{w_k\} + \min_{u_k}\left\{g_k(u_k) + b'_{k+1}f_k(u_k)\right\} + d_{k+1}
\end{aligned}$$

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, so $J_k(x_k)$ is affine. Furthermore, the optimal control at each stage solves this minimization which is independent of $x_k$. Thus the optimal policy consists of constant functions $\mu_k^*$.

### 1.28 (Monotonicity Property of DP) (www)

Under the time invariance assumptions, the DP algorithm takes the form

$$J_k(x) = \min_{u\in U(x)} \mathop{E}_{w} \left\{g(x,u,w) + J_{k+1}\big(f(x,u,w)\big)\right\}, \qquad x \in S.$$

Thus if $J_{N-1}(x) \le J_N(x)$ for all $x \in S$, we have

$$\begin{aligned}
J_{N-2}(x) &= \min_{u\in U(x)} \mathop{E}_{w} \left\{g(x,u,w) + J_{N-1}\big(f(x,u,w)\big)\right\} \\
&\le \min_{u\in U(x)} \mathop{E}_{w} \left\{g(x,u,w) + J_{N}\big(f(x,u,w)\big)\right\} \\
&= J_{N-1}(x),
\end{aligned}$$

for all $x \in S$. Continuing similarly, we obtain $J_k(x) \le J_{k+1}(x)$ for all $k$ and $x$.

# *Solutions Vol. I, Chapter 2*

**2.8 (Shortest Path Tour Problem [BeC04], [FGL13])** (www)

We will transform the problem to a (standard) shortest path problem in an expanded graph that is constructed from the given graph. Let $I$ be the set of nodes of the original graph. The expanded graph has nodes $(i,0),(i,1),\ldots,(i,N)$, where $i$ ranges over the node set $I$ of the original graph. The meaning of being in node $(i,m)$, $m=1,\ldots,N$, is that we are at node $i$ and have already successively visited the sets $T_1,\ldots,T_m$, but not the sets $T_{m+1},\ldots,T_N$. The meaning of being in node $(i,0)$ is that we are at node $i$ and have not yet visited any node in the set $T_1$.

The arcs of the expanded graph are constructed as follows: For each arc $(i,j)$ of the original graph, with length $a_{ij}$, introduce for $m=0,\ldots,N-1$, in the expanded graph an arc of length $a_{ij}$ that goes from $(i,m)$ to $(j,m)$ if $j\notin T_{m+1}$, and goes from $(i,m)$ to $(j,m+1)$ if $j\in T_{m+1}$. Also, for each arc $(i,j)$ of length $a_{ij}$ of the original graph, introduce in the expanded graph an arc of length $a_{ij}$ that goes from $(i,N)$ to $(j,N)$.

It is seen that the problem is equivalent to finding a shortest path from $(s,0)$ to $(t,N)$ in the expanded graph.

Let $D^{k+1}(i,m)$, $m=0,1,\ldots,N$, be the shortest distance from $(i,m)$ to the destination $(t,N)$ using $k$ arcs or less. The DP iteration is

$$D^{k+1}(i,m)=\min_{\{j\,\mid\,(i,j)\text{ is an arc}\}}\left\{\min_{j\notin T_{m+1}}\big\{a_{ij}+D^k(j,m)\big\},\ \min_{j\in T_{m+1}}\big\{a_{ij}+D^k(j,m+1)\big\}\right\},$$
$$m=0,\ldots,N-1,$$

$$D^{k+1}(i,N)=\begin{cases}\min_{\{j\,\mid\,(i,j)\text{ is an arc}\}}\big\{a_{ij}+D^k(j,N)\big\} & \text{if } i\neq t,\\ 0 & \text{if } i=t.\end{cases}$$

The initial condition is

$$D^0(i,m)=\begin{cases}\infty & \text{if } (i,m)\neq(t,N),\\ 0 & \text{if } (i,m)=(t,N).\end{cases}$$

This algorithm is not very efficient because it requires as many as $N\cdot|I|$ iterations, where $|I|$ is the number of nodes in the original graph. The algorithm can be made more efficient by observing that to calculate $D(i,k)$ for all $i$, we do not need to know $D(i,k-1),\ldots,D(i,0)$; it is sufficient to know just $D(j,k{+}1)$ for $j\in T_{k+1}$. Thus, we may calculate first $D(i,N)$ using a standard shortest path computation, then calculate $D(i,N-1)$, then $D(i,N-2)$, etc. This more efficient calculation process may also be viewed as a DP algorithm that involves the solution of $N$ (standard) shortest path problems involving several origins and a single destination. The origins are the nodes in $T_k$ and the destination is an artificial node to which the nodes $j\in T_{k+1}$ are connected with an arc of length $D(j,k+1)$.

**2.10 (Label Correcting with Multiple Destinations)** www

**Proposition:** If there exist 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 $\infty$.

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 follws 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 $t$.

**2.11 (Label Correcting with Negative Arc Lengths)** www

(a) The proof of termination for this algorithm follows exactly that for the original algorithm. Each time a node $j$ enters the OPEN list, its label is decreased and becomes equal to the length of some path from $s$ to $j$. Although arc lengths are no longer necessarily nonnegative, cycles lengths are. Therefore, since each path can be decomposed into a path with no repeated nodes (there is a finite number of distinct such paths) plus a (possibly empty) set of cycles (which have a nonnegative length), the number of distinct lengths of paths from $s$ to $j$ that are smaller than any given number is finite. Therefore, there can be only a finite number of label reductions and the algorithm must terminate.

(b) In the case where all arcs have nonnegative lengths, an underestimate of the shortest distance from any node to the destination node is clearly 0. Letting $u_j = 0$ from all nodes $j$, we see that the algorithm described reduces to the algorithm of Section 2.3.1.

### 2.12 (Dijkstra's Algorithm for Shortest Paths) (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 $b_0 \le b_1 \le \cdots \le b_k$. Indeed, $b_0 = 0$ and $b_1 = \min_j\{a_{s_j}\} \ge 0$, which implies that $b_0 \le b_1$. Next, we assume that $b_0 \le \cdots \le b_k$ for some $k \ge 1$; we shall prove that $b_k \le b_{k+1}$. 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,

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

Moreover,

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

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

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 $\ell^*$th iteration (with $\ell^* > k^*$), then we have $d_i^{\ell^1-1} > d_i^{\ell^*} = d_{j_\ell^*}^{\ell^*-1} + a_{j_{\ell^*}i} \ge d_{j_{\ell^*}}^{\ell^*-1} = b_{\ell^*-1}$. On the other hand, since $d_i$ is *non-increasing*, we have $b_{k^*-1} = d_i^{k^*-1} \ge d_i^{\ell^*-1}$. 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 $k$th 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 $j_{k+1}$. If the latter case applies, then the length of this path is at least the length of the shortest path from $s$ to $j^*$ through $C_k$; by the induction hypothesis, this equals $d_{j^*}^k$, which is at least $d_{j_{k+1}}^k$. It follows that, for node $j_{k+1}$ exiting the OPEN list, $d_{j_{k+1}}^k$ 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

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

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 $d_i^{k+1} = d_{j_{k+1}}^k + a_{j_{k+1}i}$. 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, $d_i^{k+1} = \min\{d_1^k, d_{j_{k+1}}^k + a_{j_{k+1}i}\}$ 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 $O(N^2)$.

**2.17 (Distributed Asynchronous Shortest Path Computation [Ber82a])** 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 $\ell$ 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

* Strictly speaking, this is the shortest distance from $s$ *to* these nodes because paths are directed from $s$ to the nodes.

$\kappa \geq k$. 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 $T \geq T_k$

**Proof:** The proof follows by induction. For $k = 0$ the proposition is true, given the positive arc length assumption. Asume 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

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

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 $d_i^T = d_i^*\ \forall\ T \geq T^*, i$.

# *Solutions Vol. I, Chapter 3*

**3.10 (Inventory Control with Integer Constraints [Vei65], [Tsi84b])** www

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

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

where

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

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 that

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

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, we have $u^{y+\delta} = u^y$ for $\delta$ sufficiently small. Thus, with $\delta$ 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

$$u^{x+1} \le u^x \le u^{x+1} + 1.$$

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

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

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

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

Let $S_k$ be the smallest real number satisfying

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

We show that $S_k$ is well defined If no $S_k$ satisfying (1) exists, we must have either $J_k(x,0) - J_k(x+1,0) > c, \;\; \forall \; x \;\in\; \Re$ or $J_k(x,0) - J_k(x+1,0) < 0, \;\; \forall \; x \;\in\; \Re$, 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 $x \ge S_k$. Using the fact that $J_k(x,u) - J_k(x+1,u)$ is nondecreasing function of $x$ we have for any $u \;\in\; \{0,1,2,\ldots\}$

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

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

### 3.18 (Optimal Termination of Sampling) (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 $P(w_k \mid x_k, u_k) \triangleq P(w_k)$ 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 $k \in S_N \triangleq \left\{ i \;\middle|\; \left(\frac{1}{N-1} + \cdots + \frac{1}{i}\right) \leq 1 \right\}$, 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 $\mu^*_{N-1}(0) = \text{continue}$, while

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

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

Assume the conclusion holds 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, then

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

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 $k \notin S_N$, then

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

where $m$ is the minimum element of $S_N$.

**Proof:** For $k = m-1$, we have

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

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

and $\mu^*_{m-1}(0) = \mu^*_{m-1}(1) = \text{continue}$. Assume the conclusion holds 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 $\mu^*_{k-1}(0) = \text{continue}$.

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

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

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

### 3.24 (Hardy's Theorem) (www)

Consider first the problem of maximizing $\sum_{i=1}^{n} a_i b_{j_i}$. We can compare the optimal sequence of $j_i$'s

$$L = \{j_1, j_2, \ldots, j_{k-1}, j, \ell, i_{k+2}, \ldots, j_n\}$$

with the sequence obtained from $L$ by interchanging $j$ and $\ell$

$$L' = \{j_1, j_2, \ldots, j_{k-1}, \ell, j, i_{k+2}, \ldots, j_n\}$$

Since this is a deterministic problem, we have

$$\begin{aligned} Reward\ of\ L &= Reward\ of\ (j_1, \ldots, j_{k-1}) + a_k b_j + a_{k+1} b_\ell \\ &\quad + Reward\ of\ (j_{k+2}, \ldots, j_n) \\ Reward\ of\ L' &= Reward\ of\ (j_1, \ldots, j_{k-1}) + a_k b_\ell + a_{k+1} b_j \\ &\quad + Reward\ of\ (j_{k+2}, \ldots, j_n). \end{aligned}$$

Since $L$ maximizes the reward, we have

$$\begin{aligned} & Reward\ of\ L \geq Reward\ of\ L' \\ \Rightarrow \quad & a_k b_j + a_{k+1} b_\ell \geq a_k b_\ell + a_{k+1} b_j \\ \Rightarrow \quad & (a_k - a_{k+1})\, b_j \geq (a_k - a_{k+1})\, b_\ell \end{aligned}$$

Since $\{a_i\}$ is a non-decreasing sequence, $a_k - a_{k+1} \leq 0$. Thus, to maximize the reward we want

$$b_j \leq b_\ell.$$

But $\{b_i\}$ is also a non-decreasing sequence. Thus, the constraint from the interchange argument can be met by setting

$$j_i = i \qquad for\ all\ i$$

To minimize $\sum_{i=1}^n a_i b_{j_i}$ we want

$$\begin{aligned} & Cost\ of\ L \leq Cost\ of\ L' \\ \Rightarrow \quad & b_j \geq b_\ell \\ \Rightarrow \quad & j_i = n - i + 1 \end{aligned}$$

since the cost of $L$ is the same as the reward for $L$ in the maximization problem.

**3.31 (Reachability of Ellipsoidal Tubes [Ber71], [BeR71a], [Ber72a])** 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

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

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

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

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

$$x' \Xi x \leq 1$$

it is sufficient that $x$ is such that

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

or by carryibf out explicitly the quadratic minimization above,

$$x' K x \leq 1,$$

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 4*

**4.1 Linear Quadratic Problems - Correlated Disturbances)** www

Define

$$y_N = x_N$$
$$y_k = x_k + A_k^{-1} w_k + A_k^{-1} A_{k+1}^{-1} w_{k+1} + \ldots + A_k^{-1} \cdots A_{N-1}^{-1} w_{N-1}$$

Then

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

and

$$y_{k+1} = A_k y_k + B_k u_k$$

Now, the cost function is the expected value of

$$x_N{}' Q x_N + \sum_{k=0}^{N-1} u_k{}' R_k u_k = y_0{}' K_0 y_0 + \sum_{k=0}^{N-1} (y_{k+1}{}' K_{k+1} y_{k+1} - y_k{}' K_k y_k + u_k{}' R_k u_k)$$

We have

$$\begin{aligned} y_{k+1}{}' K_{k+1} y_{k+1} - y_k{}' K_k y_k + u_k{}' R_k u_k &= (A_k y_k + B_k u_k)' K_{k+1} (A_k y_k + B_k u_k) + u_k{}' R_k u_k \\ &\quad - y_k{}' A_k{}' [K_{k+1} - K_{k+1} B_k (B_k{}' K_{k+1} B_k)^{-1} B_k \tau r K_{k+1}] A_k y_k \\ &= y_k{}' A_k{}' K_{k+1} A_k y_k + 2 y_k' A_k' K_{k+1} B_k u_k + u_k{}' B_k{}' K_{k+1} B_k u_k \\ &\quad - y_k{}' A_k{}' K_{k+1} A_k y_k + y_k{}' A_k{}' K_{k+1} B_k P_k^{-1} B_k' K_{k+1} A_k y_k \\ &\quad + u_k{}' R_k u_k \\ &= -2 y_k' L_k' P_k u_k + u_k{}' P_k u_k + y_k{}' L_k{}' P_k L_k y_k \\ &= (u_k - L_k y_k)' P_k (u_k - L_k y_k) \end{aligned}$$

Thus, the cost function can be written as

$$E\Big\{ y_0{}' K_0 y_0 + \sum_{k=0}^{N-1} (u_k - L_k y_k)' P_k (u_k - L_k y_k) \Big\}$$

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

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

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

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

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 $E_z\{(u-z)'P(u-z)|I\}$ for any random variable $z$, any conditioning variable $I$, and any positive semidefinite matrix $P$.) The minimization over $\mu_{N-2}$ involves

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

However, as in Lemma 4.2.1, 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 $\mu_{N-2}$ 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

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

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

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

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

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

**4.3** (www)

The control at time $k$ is $(u_k, \alpha_k)$ where $\alpha_k$ is a variable taking value 1 (if the next measurement at time $k+1$ is of type 1) or value 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

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

So

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

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

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 4.2 of the text, we have

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

The 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 $u_0$. Thus the minimization is indicated only with respect to $\alpha_0$ and not $u_0$. Because all stochastic variables are Gaussian, the quantity in the second pracket 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

$$\begin{aligned} &\underset{z_1}{E}\Big\{\underset{x_1}{E}\big\{[x_1 - E\{x_1 \mid I_1\}]'P_1[x_1 - E\{x_1 \mid I_1\}] \bigm| I_1\big\} \Bigm| z_0, u_0, \alpha_0\Big\} \\ &\qquad = \begin{cases} Tr\left(P_1^{\frac{1}{2}} \sum_{1|1}^{1} P_1^{\frac{1}{2}}\right), & \text{if } \alpha_0 = 1 \\ Tr\left(P_1^{\frac{1}{2}} \sum_{1|1}^{2} P_1^{\frac{1}{2}}\right), & \text{if } \alpha_0 = 2 \end{cases} \end{aligned}$$

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 of type 2 otherwise.

### 4.7 (Finite-State Systems - Imperfect State Information) (www)

(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 $k = N-1$,

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

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

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

This completes the induction proof that $\bar{J}_k(\lambda P_k) = \lambda \bar{J}_k(P_k)$ for all $k$.

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

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

(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

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

where $\alpha^j_{N-1} = G_{N-1}(u^j)$ and $u^j$ is the $j$th 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 $\bar{J}_k(P_k)$,

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

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

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

as $u$ ranges over the finite set of controls, $\theta$ ranges over the set of observation vector indexes $\{1, \ldots, q\}$, and $m_{u,\theta}$ 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.

## 4.10 www

(a) The state is $(x_k, d_k)$, where $d_k$ takes the value 1 or 2 depending on whether the common distribution of the $w_k$ is $F_1$ or $F_2$. The variable $d_k$ stays constant (i.e., satisfies $d_{k+1} = d_k$ for all $k$), but is not observed perfectly. Instead, the sample demand values $w_0, w_1, \ldots$ are observed ($w_k = x_k + u_k - x_{k+1}$), and provide information regarding the value of $d_k$. In particular, given the a priori probability $q$ and the demand values $w_0, \ldots, w_{k-1}$, we can calculate the conditional probability that $w_k$ will be generated according to $F_1$.

(b) A suitable sufficient statistic is $(x_k, q_k)$, where

$$
q_k = P(d_k = 1 \mid w_0, \ldots, w_{k-1}).
$$

The conditional probability $q_k$ evolves according to

$$
q_{k+1} = \frac{q_k P(w_k \mid F_1)}{q_k P(w_k \mid F_1) + (1 - q_k) P(w_k \mid F_2)}, \qquad q_0 = q,
$$

where $P\{\cdot \mid F_i\}$ denotes probability under the distribution $F_i$, and assuming that $w_k$ can take a finite number of values under the distributions $F_1$ and $F_2$.

The initial step of the DP algorithm in terms of this sufficient statistic is

$$\begin{aligned}J_{N-1}(x_{N-1}, q_{N-1}) = \min_{u_{N-1}\geq 0} \Big[ & cu_{N-1} \\ & + q_{N-1}E\big\{h\max(0, w_{N-1} - x_{N-1} - u_{N-1}) + p\max(0, x_{N-1} + u_{N-1} - w_{N-1}) \mid F_1\big\} \\ & + (1 - q_{N-1})E\big\{h\max(0, w_{N-1} - x_{N-1} - u_{N-1}) + p\max(0, x_{N-1} + u_{N-1} - w_{N-1}) \mid F_2\big\}\Big],\end{aligned}$$

where $E\{\cdot \mid F_i\}$ denotes expected value with respect to the distribution $F_i$.

The typical step of the DP algorithm is

$$\begin{aligned}J_k(x_k, q_k) = \min_{u_k\geq 0} \Big[ & cu_k \\ & + q_kE\big\{h\max(0, w_k - x_k - u_k) + p\max(0, x_k + u_k - w_k) \\ & \qquad + J_{k+1}\big(x_k + u_k - w_k,\ \phi(q_k, w_k)\big) \mid F_1\big\} \\ & + (1 - q_k)E\big\{h\max(0, w_k - x_k - u_k) + p\max(0, x_k + u_k - w_k) \\ & \qquad + J_{k+1}\big(x_k + u_k - w_k,\ \phi(q_k, w_k)\big) \mid F_2\big\}\Big],\end{aligned}$$

where

$$\phi_k(q_k, w_k) = \frac{q_kP(w_k \mid F_1)}{q_kP(w_k \mid F_1) + (1 - q_k)P(w_k \mid F_2)}.$$

(c) It can be shown inductively, as in the text, that $J_k(x_k, q_k)$ is convex and coercive as a function of $x_k$ for fixed $q_k$. For a fixed value of $q_k$, the minimization in the right-hand side of the DP minimization is exactly the same as in the text with the probability distribution of $w_k$ being the mixture of the distributions $F_1$ and $F_2$ with corresponding probabilities $q_k$ and $(1 - q_k)$. It follows that for each value of $q_k$, there is a threshold $S_k(q_k)$ such that it is optimal to order an amount $S_k(q_k) - x_k$, if $S_k(q_k) > x_k$, and to order nothing otherwise. In particular, $S_k(q_k)$ minimizes over $y$ the function

$$\begin{aligned}& cy + q_kE\big\{h\max(0, w_k - y) + p\max(0, y - w_k) + J_{k+1}\big(y - w_k,\ \phi_k(q_k, w_k)\big) \mid F_1\big\} \\ & \quad + (1 - q_k)E\big\{h\max(0, w_k - y) + p\max(0, y - w_k) + J_{k+1}\big(y - w_k,\ \phi_k(q_k, w_k)\big) \mid F_2\big\}.\end{aligned}$$

# Solutions Vol. I, Chapter 5

**5.8** (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. 5.4.1(c), this cost function is the unique solution of the system of equations

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

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

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

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

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

or equivalently

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

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

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

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

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.

**5.12** www

Let Assumption 5.2.1 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 $\pi$ and $i$, we are done. Otherwise, we must have for some $\pi$ 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 $\mu$ 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 $\mu$, and contradicts Assumption 5.2.1.

# *Solutions Vol. I, Chapter 6*

**6.8 (One-Step Lookahead Error Bound for Discretized Convex Problems)** www

We have that $\tilde{J}_N$ is the convex function $g_N$, so according to the assumption, $\hat{J}_{N-1}$ is convex. The function $\tilde{J}_{N-1}$ is the inner linearization of the convex function $\hat{J}_{N-1}$, so it is also convex. Moreover by the definition of $\tilde{J}_{N-1}$ and the convexity of $\hat{J}_{N-1}$, we have $\hat{J}_{N-1}(x) \leq \tilde{J}_{N-1}(x)$ for all $x$. From Prop. 6.1.1, it follows that

$$\overline{J}_{N-1}(x) \leq \hat{J}_{N-1}(x) \leq \tilde{J}_{N-1}(x).$$

Thus, the desired bound holds for $k = N-1$, and it is similarly proved for all $k$.

**6.9 (One-Step Lookahead Error Bound for Problem Approximation)** www

(a) We have that $\tilde{J}_k(x_k)$ is the optimal cost-to-go when the costs-per-stage and control constraint sets are $\tilde{g}_k = (x_k, u_k, w_k)$ and $\tilde{U}_k(x_k)$, respectively. We show that $\hat{J}_k(x_k)$, given by

$$\hat{J}_k(x_k) = \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\},$$

and $\hat{J}_N(x_N) = g_N(x_N)$ satisfies the assumptions of Prop. 6.1.1.

We have $\hat{J}_N(x_N) = g_N(x_N) \leq \tilde{g}_N(x_N) = \tilde{J}_N(x_N)$ for all $x_N$. For $k = 0, 1, \ldots, N-1$, we have

$$\begin{aligned}
\hat{J}_k(x_k) &= \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&\leq \min_{u_k \in \tilde{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&\leq \min_{u_k \in \tilde{U}_k(x_k)} E\left\{\tilde{g}_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&= \tilde{J}_k(x_k).
\end{aligned}$$

The first inequality follows from $\tilde{U}_k(x_k) \subset \bar{U}_k(x_k)$ for all $x_k$ and $k$, the second inequality follows from $g_k(x_k, u_k, w_k) \leq \tilde{g}_k(x_k, u_k, w_k)$ for all $x_k, u_k, w_k$ and $k$, and the last equality is the DP algorithm. Because $\hat{J}_k(x_k)$ satisfies the assumptions of Prop. 6.1.1, we have $\bar{J}_k(x_k) \leq \hat{J}_k(x_k) \leq \tilde{J}_k(x_k)$.

(b) Using the same reasoning as for part (a), we have:

$$\hat{J}_N(x_N) = g_N(x_N) \leq \tilde{g}_N(x_N) + \delta_N = \tilde{J}_N(x_N) + \delta_N$$

and

$$\begin{aligned}
\hat{J}_k(x_k) &= \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&\leq \min_{u_k \in \tilde{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&\leq \min_{u_k \in \tilde{U}_k(x_k)} E\left\{\tilde{g}_k(x_k, u_k, w_k) + \delta_k + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&= \tilde{J}_k(x_k) + \delta_k.
\end{aligned}$$

Let

$$J_k^+(x_k) = \tilde{J}_k(x_k) + \delta_k + \ldots + \delta_N.$$

Adding $\delta_{k+1} + \ldots + \delta_N$ to both sides of the second inequality above, we rewrite both inequalities above in terms of $J_k^+(x_k)$:

$$\hat{J}_N(x_N) \leq J_N^+(x_N),$$

and

$$\min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \underbrace{\tilde{J}_{k+1}\big(f_k(x_k, u_k, w_k)\big) + \delta_{k+1} + \ldots + \delta_N}_{J_{k+1}^+(f_k(x_k,u_k,w_k))}\right\} \leq \underbrace{\tilde{J}_k(x_k) + \delta_k + \delta_{k+1} + \ldots + \delta_N}_{J_k^+(x_k)}$$

Letting

$$\hat{J}_k^+(x_k) = \min_{u_k \in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + J_{k+1}^+(f_k(x_k, u_k, w_k))\right\}$$

and $\hat{J}_N^+ = \hat{J}_N = g_N$, we rewrite the second inequality above as $\hat{J}_k^+(x_k) \leq J_k^+(x_k)$. If we use $J_k^+(x_k)$ in place of $\tilde{J}_k(x_k)$ as a cost-to-go approximation, the one-step lookahead policy will remain unchanged (since $J_k^+$ and $\tilde{J}_k$ differ by the same constant for all states), and we can apply Prop. 6.1.1 to obtain $\bar{J}_k(x_k) \leq \hat{J}_k^+(x_k) \leq J_k^+(x_k)$. It follows that

$$\bar{J}_k(x_k) \leq \tilde{J}_k(x_k) + \delta_k + \ldots + \delta_N.$$

### 6.11 (One-Step Lookahead/Rollout for Shortest Paths) 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 $\overline{P}_i$ is unique since there is only one arc outgoing from each node.

Let $\overline{P}_i = (i, i_1, i_2, \ldots, i_k, n)$ [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

$$\hat{F}(i_1) = a_{i_1i_2} + F(i_2) \geq a_{i_1i_2} + \hat{F}(i_2),$$

$$\cdots$$

$$\hat{F}(i_k) = a_{i_kn} + F(i_n) = a_{i_kn}.$$

By adding the above relations, we obtain

$$\hat{F}(i) \geq a_{ii_1} + a_{i_1i_2} + \cdots + a_{i_kn}.$$

The result follows since the right-hand side is the length of $\overline{P}_i$.

We may view the set of arcs $\overline{P}_i$ as a one-step lookahead policy with lookahead function equal to $F(i)$, and with the scalars $\hat{F}(i)$ identified as the function $\hat{J}_k$ of Prop. 6.1.1. The conclusion of that proposition states that the cost function of the one-step lookahead policy, which is defined by the lengths of the paths $\overline{P}_i$, is majorized by $\hat{J}_k$, which can be identified with the scalars $\hat{F}(i)$. This is exactly what we have shown.

(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

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

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

$$F(i) \geq \min_{j \in \mathcal{N}_i} \big[a_{ij} + F(j)\big],$$

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.

**Exercise 6.12 (Performance Bounds for Two-Step Lookahead Policies)** www

First note that by definition, $J_N^+(x_N) = \hat{J}_N(x_N) = \tilde{J}_N(x_N) = g_N(x_N)$, so the conclusion holds for $k = N$. For $k = N-1$ we have

$$\begin{aligned}
J_{N-1}^+(x_{N-1}) &= \min_{u_{N-1}\in \bar{U}_{N-1}(x_{N-1})} E\big\{g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + \hat{J}_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}))\big\} \\
&= \min_{u_{N-1}\in \bar{U}_{N-1}(x_{N-1})} E\big\{g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + \tilde{J}_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}))\big\} \\
&= \hat{J}_{N-1}(x_{N-1}).
\end{aligned}$$

Proceeding by induction, for $k < N-1$, assuming that $\hat{J}_{k+1} \leq \tilde{J}_{k+1}$, we have

$$\begin{aligned}
J_k^+(x_k) &= \min_{u_k\in \bar{U}_k(x_k)} E\big\{g_k(x_k, u_k, w_k) + \hat{J}_{k+1}(f_k(x_k, u_k, w_k))\big\} \\
&\leq \min_{u_k\in \bar{U}_k(x_k)} E\big\{g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}(f_k(x_k, u_k, w_k))\big\} \\
&= \hat{J}_k(x_k).
\end{aligned}$$

This proves that $J_k^+(x_k) \leq \hat{J}_k(x_k)$ for all $k = 0, \ldots, N-1$.

There remains to show that $\bar{J}_k(x_k) \leq \hat{J}_k^+(x_k)$ for $k = 0, \ldots, N-1$. Indeed, for $k = N-1$ we have [recall that $\hat{J}_N(x_N) = \bar{J}_N(x_N)$]

$$\begin{aligned}
\bar{J}_{N-1}(x_{N-1}) &= E\big\{g_{N-1}(x_{N-1}, \bar{\mu}_{N-1}(x_{N-1}), w_{N-1}) + \bar{J}_N(f_{N-1}(x_{N-1}, \bar{\mu}_{N-1}(x_{N-1}), w_{N-1}))\big\} \\
&= E\big\{g_{N-1}(x_{N-1}, \bar{\mu}_{N-1}(x_{N-1}), w_{N-1}) + \hat{J}_N(f_{N-1}(x_{N-1}, \bar{\mu}_{N-1}(x_{N-1}), w_{N-1}))\big\},
\end{aligned}$$

where $\bar{\mu}_{N-1}(x_{N-1})$ attains the minimum in the expression

$$\min_{u_{N-1}\in \bar{U}_{N-1}(x_{N-1})} E\big\{g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + \hat{J}_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}))\big\}.$$

Therefore we have

$$\begin{aligned}
\bar{J}_{N-1}(x_{N-1}) &= \min_{u_{N-1}\in \bar{U}_{N-1}(x_{N-1})} E\big\{g_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}) + \hat{J}_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}))\big\} \\
&= J_{N-1}^+(x_{N-1}).
\end{aligned}$$

Proceeding by induction, for $k < N-1$, assuming that $\bar{J}_{k+1} \leq J_{k+1}^+$, we have

$$\begin{aligned}
\bar{J}_k(x_k) &= E\big\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \bar{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\big\} \\
&\leq E\big\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + J_{k+1}^+(f_k(x_k, \bar{\mu}_k(x_k), w_k))\big\} \\
&\leq E\big\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \hat{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\big\},
\end{aligned}$$

where $\bar{\mu}_k(x_k)$ attains the minimum in the expression

$$\min_{u_k\in \bar{U}_k(x_k)} E\big\{g_k(x_k, u_k, w_k) + \hat{J}_{k+1}(f_k(x_k, u_k, w_k))\big\},$$

and $\hat{J}_{k+1}$ is obtained itself on the basis of a one-step lookahead approximation:

$$\hat{J}_{k+1}(x_{k+1}) = \min_{u_{k+1}\in \bar{U}_{k+1}(x_{k+1})} E\left\{g_{k+1}(x_{k+1}, u_{k+1}, w_{k+1}) + \tilde{J}_{k+2}(f_{k+1}(x_{k+1}, u_{k+1}, w_{k+1}))\right\},$$

with $x_{k+1} = f_k(x_k, u_k, w_k)$. Then, using the induction hypothesis $\bar{J}_{k+1} \leq J^+_{k+1}$ and the fact $J^+_{k+1} \leq \hat{J}_{k+1}$ shown earlier, we have

$$\begin{aligned}
\bar{J}_k(x_k) &= E\left\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \bar{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\right\} \\
&\leq E\left\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + J^+_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\right\} \\
&\leq E\left\{g_k(x_k, \bar{\mu}_k(x_k), w_k) + \hat{J}_{k+1}(f_k(x_k, \bar{\mu}_k(x_k), w_k))\right\} \\
&= \min_{u_k\in \bar{U}_k(x_k)} E\left\{g_k(x_k, u_k, w_k) + \hat{J}_{k+1}(f_k(x_k, u_k, w_k))\right\} \\
&= J^+_k(x_k).
\end{aligned}$$

The induction is complete.

### 6.13 (Rollout Algorithms with Errors) www

(a) Assuming $|e(j)| = |\hat{H}(j) - H(j)| \leq \epsilon$ for all $j$, we have for $m = 1, \ldots, \bar{m} - 1$,

$$H(i_{m+1}) - \epsilon \leq \hat{H}(i_{m+1}) = \min_{j\in N(i_m)} \hat{H}(j) \leq \min_{j\in N(i_m)} H(j) + \epsilon \leq H(i_m) + \epsilon,$$

where the first and second inequalities follow from the bounds on $e(j)$, the equality follows from the definition of the rollout algorithm, and the last inequality follows from the definition of sequential improvement. We use the above relation, $H(i_{m+1}) - \epsilon \leq H(i_m) + \epsilon$, to obtain a bound on $H(i_{\bar{m}})$, the cost of the generated path, in terms of $H(i_1)$:

$$H(i_{\bar{m}}) \leq H(i_{\bar{m}-1}) + 2\epsilon \leq H(i_{\bar{m}-2}) + 4\epsilon \leq \cdots \leq H(i_1) + 2(\bar{m} - 1)\epsilon.$$

(b) Assuming $0 \leq \hat{H}(j) - H(j) \leq \epsilon$ for all $j$, we have for $m = 1, 2, \ldots, \bar{m} - 1$,

$$H(i_{m+1}) \leq \hat{H}(i_{m+1}) = \min_{j\in N(i_m)} \hat{H}(j) \leq \min_{j\in N(i_m)} H(j) + \epsilon \leq H(i_m) + \epsilon.$$

We use the above relation, $H(i_{m+1}) \leq H(i_m) + \epsilon$, to obtain a bound on $H(i_{\bar{m}})$, the cost of the generated path, in terms of $H(i_1)$:

$$H(i_{\bar{m}}) \leq H(i_{\bar{m}-1}) + \epsilon \leq H(i_{\bar{m}-2}) + 2\epsilon \leq \cdots \leq H(i_1) + (\bar{m} - 1)\epsilon.$$

Assuming $-\epsilon \leq \hat{H}(j) - H(j) \leq 0$ for all $j$, we have for $m = 1, 2, \ldots, \bar{m} - 1$,

$$H(i_{m+1}) - \epsilon \leq \hat{H}(i_{m+1}) = \min_{j\in N(i_m)} \hat{H}(j) \leq \min_{j\in N(i_m)} H(j) \leq H(i_m).$$

We use the above relation, $H(i_{m+1}) - \epsilon \leq H(i_m)$, to find a bound on $\hat{H}(i_{\bar{m}})$, the cost of the generated path, in terms of $H(i_1)$:

$$H(i_{\bar{m}}) \leq H(i_{\bar{m}-1}) + \epsilon \leq H(i_{\bar{m}-2}) + 2\epsilon \leq \cdots \leq H(i_1) + (\bar{m}-1)\epsilon.$$

(c) If the base heuristic is optimal, then $H(i_1)$ is the optimal cost starting from $i_1$. We have the following bound on the difference between the cost of the generated path and the optimal cost starting from $i_1$:

$$H(i_{\bar{m}}) - H(i_1) \leq 2(\bar{m}-1) \max_j \left| e(j) \right|$$

Thus, if we use a DP algorithm, or any other method, to calculate the optimal cost-to-go with an error of at most $\epsilon$, and then use the calculated values to generate a path/solution, the cost of this solution will be within $2(\bar{m}-1)\epsilon$ of the optimal cost.

# *Solutions Vol. I, Chapter 7*

### 7.6 (L'Hôpital's Problem) www

This problem is similar to the Brachistochrone Problem described in the text (Example 7.4.2). 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

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

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

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

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

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

Thus an optimal trajectory satisfies the differential equation

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

It can be seen through straightforward calculation that the curve

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

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

**7.9** (www)

We have the system $\dot{x}(t) = Ax(t) + Bu(t)$, and the quadratic cost

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

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

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

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

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

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. $(*)$, we have

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

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.