6.231 Dynamic Programming and Stochastic Control
Fall 2008

6.231 Dynamic Programming

Midterm Exam, Fall 2002

October 29, 2002 2:30-4:30pm

Prof. Dimitri Bertsekas

Problem 1 (20 points)

We have a set of N objects, denoted $1, 2, \ldots, N$, which we want to group in clusters that consist of consecutive objects. For each cluster $i, i{+}1, \ldots, j$, there is an associated cost a_{ij}. We want to find a grouping of the objects in clusters such that the total cost is minimum. Formulate the problem as a shortest path problem, and write a DP algorithm for its solution. (Note: An example of this problem arises in typesetting programs, such as TEX/LATEX, that break down a paragraph into lines in a way that optimizes the paragraph's appearance.)

Problem 2 (40 points)

The latest casino sensation is a slot machine with N arms, labeled $1, \ldots, N$. A single play with arm i costs C_i dollars, and has two possible outcomes: a "win," which occurs with probability p_i and pays a reward R_i, and a "loss," which occurs with probability $1 - p_i$. The rule is that each arm may be played at most once, and play must stop at the first loss or after playing all arms once, whichever comes first. The objective is to find the arm-playing order that maximizes the total expected reward minus the total expected cost.

(a) Write a DP algorithm for solving the problem.

(b) Show that it is optimal to play the arms in order of nonincreasing $(p_iR_i - C_i)/(1 - p_i)$. *Note*: This may be shown with or without using the DP algorithm of part (a).

(c) Assume that at any time, there is the option to stop playing, in addition to selecting a new arm to play. Write a DP algorithm for solving this variant of the problem, and find an optimal policy.

(d) Suppose that in the context of part (c), you may play an arm as many times as you want, but each time the reward to be obtained diminishes by a factor β with $0 < \beta < 1$. Assuming that $C_i > 0$, find an optimal policy.

Problem 3 (40 points)

Consider an inventory control problem where stock evolves according to

$$x_{k+1} = x_k + u_k - w_k,$$

and the cost of stage k is

$$cu_k + h \max(0, w_k - x_k - u_k) + p \max(0, x_k + u_k - w_k),$$

where c, h, and p are positive scalars with $p > c$. There is no terminal cost. The stock x_k is perfectly observed at each stage. The demands w_k are independent, identically distributed, nonnegative random variables. However, the (common) distribution of the w_k is unknown. Instead it is known that this distribution is one out of two known distributions F_1 and F_2, and that the a priori probability that F_1 is the correct distribution is a given scalar q, with $0 < q < 1$. You may assume for convenience that w_k can take a finite number of values under each of F_1 and F_2.

(a) Formulate this as an imperfect state information problem, and identify the state, control, system disturbance, observation, and observation disturbance.

(b) Write a DP algorithm in terms of a suitable sufficient statistic.

(c) Characterize as best as you can the optimal policy.

MIDTERM SOLUTIONS:

Problem 1 (20 points)

We may model this problem as a deterministic shortest path problem with nodes $\{0, 1, \ldots, N\}$, where 0 is the start and N is the destination, and arcs (i, j) only if $j > i$ (unless you are at node N which is absorbing). So each arc (i, j), for $i \neq N$, corresponds to a cluster of nodes $i + 1, i + 2, \ldots, j$ which has cost $a_{i+1,j}$, while arc (N, N) has cost 0.

We have the following DP problem setup:

x_k=last node of a cluster

$x_k \in S = \{0, 1, \ldots, N\}$ for $k = 0, 1, \ldots, N$

$x_{k+1} = u_k$ for $k = 0, 1, \ldots, N - 1$

$x_0 = 0$

$$u_k \in U_k(x) = \{i \in S \mid i > x\} \quad if \quad x \neq N \qquad k = 0, 1, \ldots, N - 1$$

and

$$u_k \in U_k(x) = \{N\} \quad if \quad x = N$$

Moreover:

$$g_k(x,u) = a_{x+1,u} \quad if \quad x \neq N \qquad k = 0, 1 \ldots, N-1$$

and

$$g_k(x,u) = 0 \quad if \quad x = N$$

We then have the following DP algorithm:

$$J_N(N) = 0$$

$$J_k(i) = \min_{\{j \in S | j > i\}} [a_{i+1,j} + J_{k+1}(j)] \quad if \quad i \neq N \qquad k = 0, 1, \ldots, N-1$$

and

$$J_k(i) = 0 \quad if \quad i = N$$

The optimal cost is then $J_0(0)$.

Problem 2 (40 points)

(a) (12 points) Choose as state at stage k the set of $N-k$ arms not yet played (if no loss has occured in the first $k-1$ plays) or a special termination state otherwise (which has cost-to-go 0 at all stages under any policy). The initial state is then the set $\{1, 2, \ldots, N\}$. The expected reward at each stage is $p_i R_i - C_i$, where i is the arm selected to play. The DP algorithm at the nontermination states is

$$J_k\big(\{i_1, \ldots, i_{N-k}\}\big) = \max_{i \in \{1_1, \ldots, i_{N-k}\}} \big[p_i R_i - C_i + p_i J_{k+1}\big(\{i_1, \ldots, i_{N-k}\} - \{i\}\big)\big], \qquad k = 0, \ldots, N-1,$$

$$J_N(\emptyset) = 0.$$

(b) (12 points) The problem is identical to the quiz problem with expected reward for trying the ith question equal to $p_i R_i - C_i$. The result follows by the interchange argument in the book, with expected reward given the order $i_1, i_2, \ldots, i_N$ equal to $p_{i_1} R_{i_1} - C_{i_1} + p_{i_1}(p_{i_2} R_{i_2} - C_{i_2}) + \ldots + p_{i_1} p_{i_2} \cdots p_{i_{N-1}}(p_{i_N} R_{i_N} - C_{i_N})$.

(c) (8 points) Transform the problem to the one of parts (a) and (b) by adding a new arm $N+1$ with

$$C_{N+1} = 0, \qquad p_{N+1} = 0, \qquad R_{N+1} = 0.$$

Choosing this arm is equivalent to stopping, since its cost, reward, and probability of continuing are all equal to 0. Since this new arm's index $((p_i R_i - C_i)/(1-p_i))$ is equal to 0, we never play any arm with negative index. An optimal policy is to select the set of arms i with $p_i R_i > C_i$, play them in order of nonincreasing $(p_i R_i - C_i)/(1-p_i)$, and then stop.

(d) (8 points) Create additional arms with reward $\beta^{m-1}R_i$ corresponding to playing arm i for the mth time. However, create new arms only for the finite number of values of m for which $p_i\beta^{m-1}R_i > C_i$. We initially ignore the constraint that arm (i,l) must be played before arm $(i,l+1)$ for all i,l. Using the results of part (c), the optimal policy for this problem is to select the set of arms with (i,m) satisfying $p_i\beta^{m-1}R_i > C_i$, play them in order of nonincreasing $(p_i\beta^{m-1}R_i - C_i)/(1-p_i)$, and then stop. Notice that this optimal policy, for any i, always plays arm (i,l) before arm $(i,l+1)$ for all l, and therefore is also optimal over the set of orderings with this constraint.

Problem 3 (40 points)

(a) (13 points) 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) (13 points) 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

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

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

The typical step of the DP algorithm is

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

where

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

(c) (14 points) 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

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