6.231 Dynamic Programming and Stochastic Control
Fall 2008

6.231 Dynamic Programming

Midterm Exam, Fall 2001

Prof. Dimitri Bertsekas

Problem 1 (33 points)

An enterprising but somewhat foolish graduate student has invested next semester's tuition in the stock market. As a result, he currently possesses a certain amount of stock that he/she must sell by registration day, which is N days away. The stock must be sold in its entirety on a single day, and will then be deposited in a bank where it will earn interest at a daily rate r. The value of the stock on day k is denoted by x_k and it evolves according to

$$x_{k+1} = \lambda x_k + w_k, \qquad x_0 : \text{given},$$

where λ is a scalar with $0 < \lambda < 1$, and w_k is a random variable taking one of a finite number of positive values. We assume that $w_0, \ldots, w_{N-1}$ are independent and identically distributed. The student wants to maximize the expected value of the money he/she has on registration day.

(a) Write a DP algorithm to solve this problem.

(b) Analyze the DP algorithm to characterize as best as you can the optimal selling policy.

(c) Assume that the student has the option of selling only a portion of his stock on a given day. What if anything would he/she do different? (A nonmathematical argument will suffice here.)

Problem 2 (33 points)

A businessman operates out of a van that he sets up in one of two locations on each day. If he operates in location i on day k, he makes a known and predictable profit, denoted r_k^i. However, each time he moves from one location to another, he pays a setup cost c. The businessman wants to maximize his total profit over N days.

(a) Show that the problem can be formulated as a shortest path problem, and write the corresponding DP algorithm.

(b) Suppose he is at location i on day k. Let

$$R_k^i = r_k^{\bar{i}} - r_k^i,$$

where $\bar{i}$ denotes the location that is not equal to i. Show that if $R_k^i \leq 0$ it is optimal to stay at location i, while if $R_k^i \geq 2c$, it is optimal to switch.

(c) Suppose that on each day there is a probability of rain p^i at location i independently of other locations and independently of whether it rained on other days. If he is at location i and it rains, his profit for the day is reduced by a factor of β^i. Can the problem still be formulated as a shortest path problem? Write a DP algorithm.

(d) (OPTIONAL EXTRA CREDIT 10 points) Suppose there is a possibility of rain as in part (c), but the businessman receives an accurate rain forecast just before making the decision to switch or not switch locations. Can the problem still be formulated as a shortest path problem? Write a DP algorithm.

Problem 3 (34 points)

Consider the search problem of Example 5.4.1 of your text (p. 247) for different values of the search horizon N.

(a) Show that for any value of the a priori probability p_0 that is strictly less than 1, there is a threshold value of N, call it $\overline{N}$, such that the optimal reward function $J_0(p_0)$ is independent of N as long as $N > \overline{N}$.

(b) For N greater than the threshold $\overline{N}$ of part (a) and for a given value of p_0, give a method to calculate the value of $J_0(p_0)$ that does not use the DP algorithm.

(c) Suppose that there are two sites that can be searched, instead of one. The sites may contain a treasure of corresponding values V^1 and V^2 (independently of each other), and the probabilities of a successful search are β^1 and β^2, respectively. After finding a treasure in one site, one may continue searching for the treasure in the other site (but of course each search costs C). Write a DP algorithm involving the probabilities p_k^1 and p_k^2 that a treasure is present at sites 1 and 2, respectively.

(d) Under the assumptions of part (c), show that for any values of the a priori probabilities p_0^1, p_0^2, there is a threshold value of N, call it $\overline{N}$, such that the optimal cost-to-go function $J_0(p_0^1, p_0^2)$ is independent of N as long as $N > \overline{N}$. Find the optimal search policy if $N > \overline{N}$.