B9140 Dynamic Programming and Reinforcement Learning Fall 2017
Prof. Daniel Russo September 11, 2017

Homework 1, Due in class Monday September 18

When formulating a problem and/or providing a dynamic programming recursion, make sure to clearly define the state space, action space, cost function, and state dynamics. When characterizing an optimal policy, make sure to clearly define the nature of the state that it takes as input and the action that it produces as output.

1 Deterministic Costs

In class, we formulated a problem where the cost incurred in period k, $g_k(x_k, u_k, w_k)$, is a function not only of the state x_k and control u_k but of the random disturbance w_k. Consider a modified MDP with the same transition dynamics

$$x_{k+1} = f_k(x_k, u_k, w_k) \quad k \in \{0, 1, \ldots, N-1\}$$

but where costs incurred at stage k are a deterministic function of $\tilde{g}_k(x_k, u_k)$ of the state and control, defined by

$$\tilde{g}_k(x, u) = \mathbb{E}[g_k(x, u, w_k)] \quad \forall x \in \mathcal{X}_k,\ u \in U_k(x).$$

Show that the optimal cost–to–go function $J^*(x_0)$ and the optimal policy is the same for both problems. (*You may assume for simplicity that there is a unique optimal policy for the problem with random costs* $g_k(x_k, u_k, w_k)$.)

2 Optimal Sequential Search

Consider the problem of actively searching for the location of an unknown target $z^* \in [0, 1]$. At each time k, we query a location $u_k \in [0, 1]$ and are told whether z^* is smaller or larger than u_k. (We observe $\mathbf{1}\{z^* > u_k\}$) Based on these observations, we can construct increasingly refined intervals $[a_k, b_k] \subseteq [a_{k-1}, b_{k-1}] \subseteq \ldots \subseteq [0, 1]$ that are guaranteed to contain z^*. In particular, $[a_1, b_1] = [0, u_0]$ if we observe that $z^* \leq u_0$ and is $[u_0, 1]$ otherwise.

We will use dynamic programming to study how to sequentially acquire information about z^* in an optimal manner. Assume the location of the target z^* is drawn uniformly at random from $[0, 1]$. The objective is to sequentially choose the querry points $u_0, u_2, \ldots u_{N-1}$ to minimize $\mathbb{E}\left[\log(b_N - a_N)\right]$.

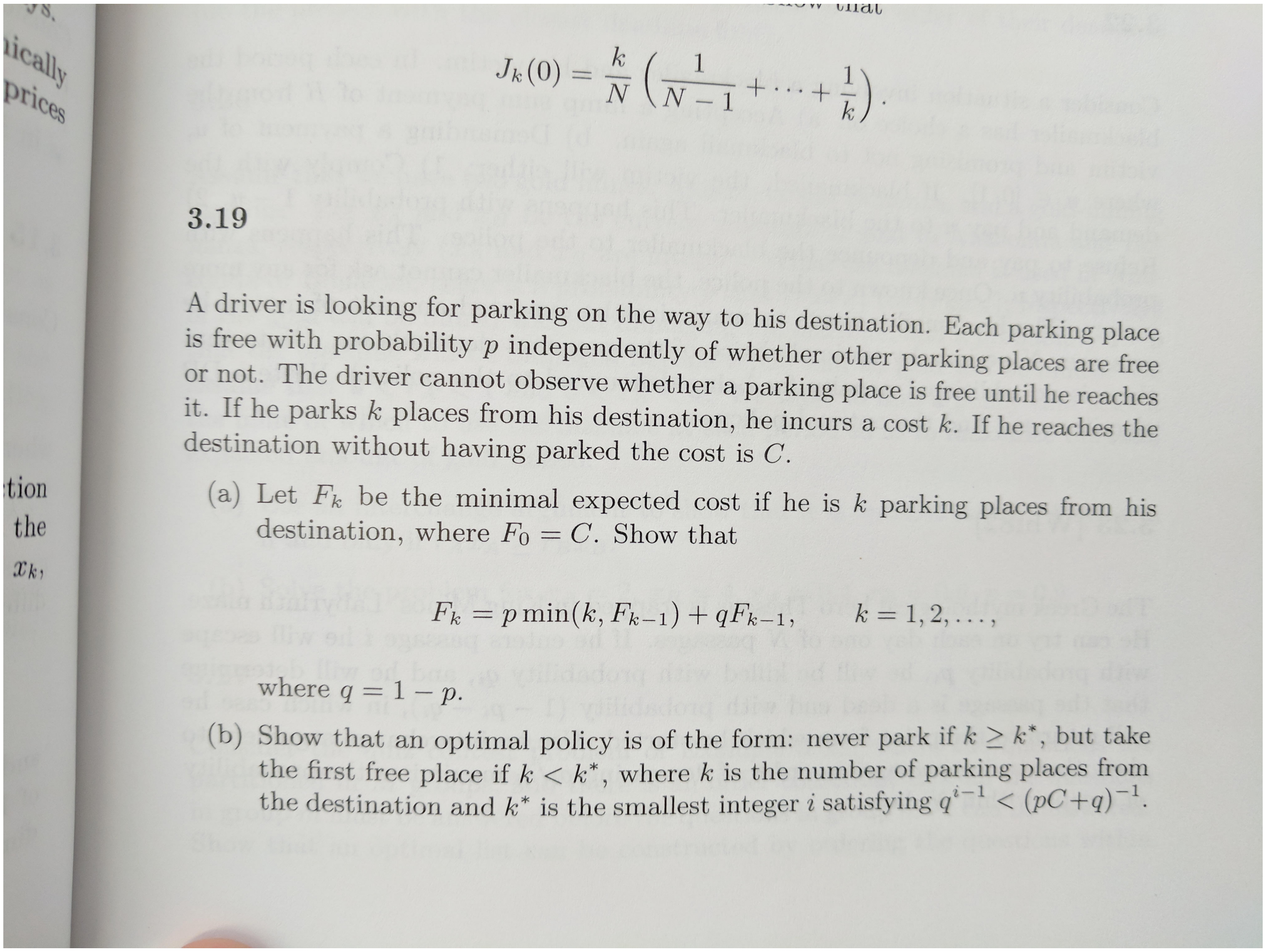
a) Formulate this problem as a finite horizon Markov decision process.

b) Solve for the optimal policy $\mu^*_{N-1}(a_{N-1}, b_{N-1})$ and cost-to-go function J^*_{N-1} at stage $N-1$. **Hint:** it is easier to work with the variable $p_k \equiv (u_k - a_k)/(b_k - a_k) \in [0,1]$

c) Prove that a myopic policy is optimal. That is, show $\mu^*_k = \mu^*_{N-1}$ for all k.

3 Optimal Stopping

Solve problem 3.19 of Bertsekas Vol. 1, reproduced below.



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

3.19

A driver is looking for parking on the way to his destination. Each parking place is free with probability p independently of whether other parking places are free or not. The driver cannot observe whether a parking place is free until he reaches it. If he parks k places from his destination, he incurs a cost k. If he reaches the destination without having parked the cost is C.

(a) Let F_k be the minimal expected cost if he is k parking places from his destination, where $F_0 = C$. Show that

$$F_k = p\min(k, F_{k-1}) + qF_{k-1}, \qquad k = 1, 2, \ldots,$$

where $q = 1 - p$.

(b) Show that an optimal policy is of the form: never park if $k \geq k^*$, but take the first free place if $k < k^*$, where k is the number of parking places from the destination and k^* is the smallest integer i satisfying $q^{i-1} < (pC+q)^{-1}$.