Chapter1 (Recommended Exercises)

Akshat Bisht

February 27, 2020

3.Randomized Ski Rental Problem

Given:
P(buy ski on the 1st day) = $\frac{b-1}{b}$; P(rent and never buy)=$\frac{1}{b}$
Cost of buying=b and Cost of renting=1
Weather spoils on day k.

1. Expected cost of algorithm in terms of b and k?

probablity	cost(ALG)
$\frac{b-1}{b}$	b
$\frac{1}{b}$	k

$$\mathbb{E}(ALG) = \frac{b-1}{b} \times b + \frac{1}{b} \times k$$
$$= b + \frac{k}{b} - 1$$

2. Does the Algorithm achieve constant competitive ratio?

There are 2 Cases for different values of OPT cost
a) $k \leq$ b —> OPT=k

$$\frac{\mathbb{E}(ALG)}{OPT} = \frac{b + \frac{k}{b} - 1}{k}$$
$$= \lim_{k \to \infty} \frac{b}{k} + \frac{1}{b} - \frac{1}{k}$$
$$= \frac{1}{b}$$

b) k>b —>OPT=b

$$\frac{\mathbb{E}(ALG)}{OPT} = \frac{b + \frac{k}{b} - 1}{b}$$

$$= \lim_{k \to \infty} 1 + \frac{k}{b^2} - \frac{1}{b}$$

$$= \infty$$

Thus there is no constant competitive ratio.

3. New Randomization Algorithm. What is the ρ?

   Given:
   Rent equipment for b-1 days. For each of the following day, buy epuipment with a chance of 1/3 and continue renting otherwise. If you decide to buy on a particular day you don't make decisions the following day.

   let k be the number of days of AFTER b-1 days that the person can ski.$k \in W$
   P(Buy)=1/3 (After b-1 days)

days(d)	probablity	cost(ALG)	cost(OPT)
$(b-1)$	1	$(b-1)$	$(b-1)$
$(b-1)+1$	$\frac{1}{3}$	$(b-1)+b$	b
$(b-1)+2$	$\frac{1}{3} \times \frac{2}{3}$	$(b-1)+(1+b)$	b
$(b-1)+3$	$\frac{1}{3} \times (\frac{2}{3})^2$	$(b-1)+(2+b)$	b
$(b-1)+4$	$\frac{1}{3} \times (\frac{2}{3})^3$	$(b-1)+(3+b)$	b
.	.	.	.
.	.	.	.
$(b-1)+k$	$\frac{1}{3} \times (\frac{2}{3})^{k-1}$	$(b-1)+(k+b)$	b

$$\mathbb{E}(ALG) = \sum_{k=0}^{\infty} ((b-1)+(k+b)) \times (\frac{2}{3})^{k-1} \times \frac{1}{3}$$

$$\mathbb{E}(ALG) = \sum_{k=0}^{\infty} (2b+k-1) \times (\frac{2}{3})^{k} \times \frac{1}{2}$$

$$\mathbb{E}(ALG) = 3b + \frac{1}{2}$$

3. Randomized Line Search Problem

P(initial direction to be +1) = $\frac{1}{2}$
P(initial direction to be -1) = $\frac{1}{2}$
Rest of the startegy remains the same.

Suppose that the object has been placed at distance $d \geq 1$ from the origin.
OPT=d.
In phase i the robot visits location $(-2)^i$ and
travels distance $(-2)^i \times 2$
The worst case is when an object is located just outside of the radius covered in some phase.
Then the robot returns to the origin, doubles the distance and travels in the “wrong direction”, returns to the origin, and discovers the object by travelling in the “right direction.”
In other words when the object is at distance $d, 2^i < d \leq 2^{i+1}$ in the direction $(-1)^i$
The total distance travelled is

$$2(1 + 2 + ... + 2^i + 2^{i+1}) + d \leq 2 \times 2^{i+2} + d < 8d + d = 9d$$

This doubling strategy gives a 9-competitive algorithm for the line search problem.

From our analysis of the deterministic algorithm, we see that we got lucky when the robot started walking in the "good" direction at the beginning of the algorithm. Since the adversary can see our algorithm, however, he always chooses to put the door in the "bad" direction. Thus, a natural way to improve the competitive ratio of our algorithm is to ip a fair coin to decide which direction to start walking in. If the adversary is oblivious, then he doesn’t know the outcome of the coin ip, and we now have an equal chance of getting lucky or unlucky.