Design and Analysis of Algorithms

NIRDOSH BHATNAGAR

Probabilistic Analysis and Randomized Algorithms

1. Introduction

This note introduces probabilistic analysis and randomized algorithms. The advantages of probabilistic analysis and randomized algorithms are illustrated via an example. We analyze the so-called hiring problem.

2. The Hiring Problem

You are using an employment agency to hire a new office assistant. The agency sends you one candidate each day. You interview the candidate and must immediately decide whether or not to hire that person. But if you hire, you must also fire your current office assistant - even if it's someone you have recently hired.

You are committed to having hired, at all times, the best candidate seen so far. Meaning that whenever you interview a candidate who is better than your current office assistant, you must fire the current office assistant and hire the candidate. Since you must have someone hired at all times, you will always hire the first candidate that you interview.

- Cost to interview is c_i per candidate. The interview fee is paid to agency.
- Cost to hire is c_h per candidate. It includes cost to fire current office assistant and the hiring fees is paid to agency.

Assume that $c_h > c_i$. The pseudocode to model this algorithm is given below. Assume that the candidates are numbered from 1 to n.

```
HIRE-ASSISTANT (n)
best = 0      // candidate 0 is a least-qualified dummy candidate
for i = 1 to n
    interview candidate i
    if candidate i is better than candidate best
        best = i
        hire candidate i
```

If out of n candidates m of them are hired, then the cost is $(nc_i + mc_h)$.

2.1. Worst-case Analysis.

In the worst-case all the n candidates are hired. Then the cost is $O(nc_i + nc_h)$ which is equal to $O(nc_h)$, since $c_h > c_i$.

2.2. Probabilistic Analysis.

Assume that the candidates arrive in random order. Let X be a random variable that equals the number of times we hire a new office assistant. Also define

$$X_i = \begin{cases} 1, & \text{if candidate } i \text{ is hired} \\ 0, & \text{if candidate } i \text{ is not hired} \end{cases} \quad \text{where } 1 \leq i \leq n$$

Thus $X = \sum_{i=1}^{n} X_i$. Let $\mathcal{E}(\cdot)$ denote the expectation operator. Thus $\mathcal{E}(X_i)$ is the probability that the candidate i is hired. If the candidates are ranked randomly, then the probability that candidate i is the best so far is equal to $1/i$. Thus $\mathcal{E}(X_i) = 1/i$. Therefore

$$\mathcal{E}(X) = \sum_{i=1}^{n} \frac{1}{i} = \ln(n) + O(1)$$

Thus the expected (average-case) hiring cost is $O(nc_i + c_h \ln(n))$.

2.3. Randomized Algorithm. Instead of assuming a distribution of the inputs, we impose a distribution. In the hiring problem, instead of interviewing the candidates in the order presented, we randomly permute the order. The randomization is now in the algorithm, and not in the input distribution. A pseudocode for randomized hire-assistant is given below.

```
RANDOMIZED HIRE-ASSISTANT (n)
randomly permute the list of candidates.
best = 0     // candidate 0 is a least-qualified dummy candidate
for i = 1 to n
    interview candidate i
    if candidate i is better than candidate best
        best = i
        hire candidate i
```

Note that the only change in the algorithm, is the initial random permutation of the list of candidates. With this change, the expected hiring cost is equal to that obtained via a probabilistic analysis of the original HIRE-ASSISTANT algorithm, which is $O(nc_i + c_h \ln(n))$.