The Design and Analysis of Algorithms

Lecture 26 Randomized Algorithms I

Zhenbo Wang

Department of Mathematical Sciences, Tsinghua University

Content

Randomization

- *Algorithmic design patterns*.

  Greedy.

  Divide-and-conquer.

  Dynamic programming.

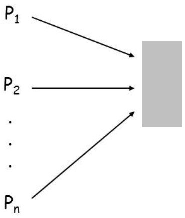
  Randomization: Allow fair coin flip in unit time.

- *Why randomize?* Can lead to simplest, fastest, or only known algorithm for a particular problem.

R. Motwani and P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.

Contention Resolution in a Distributed System

- *Contention resolution*. Given n processes $P_1, \cdots, P_n$, each competing for access to a shared database. If two or more processes access the database simultaneously, all processes are locked out. Devise protocol to ensure all processes get through on a regular basis.
- *Restriction*. Processes can't communicate.
- *Challenge*. Need symmetry-breaking paradigm.



Contention Resolution: Randomized Protocol

- *Protocol*. Each process requests access to the database at time t with probability $p = 1/n$.

Claim. Let $S[i, t]$ = event that process i succeeds in accessing the database at time t. Then $1/(e \cdot n) \leq \Pr[S(i, t)] \leq 1/(2n)$.

Pf. By independence, $\Pr[S(i, t)] = p(1 - p)^{n-1}$.

Setting $p = 1/n$, we have $\Pr[S(i, t)] = 1/n(1 - 1/n)^{n-1}$. $\square$

- *Useful facts from calculus*. As n increases from 2, the function:

$(1 - 1/n)^n$ converges monotonically from $1/4$ up to $1/e$.

$(1 - 1/n)^{n-1}$ converges monotonically from $1/2$ down to $1/e$.

Contention Resolution: Randomized Protocol

Claim. The probability that process i fails to access the database in en rounds is at most $1/e$. After $e \cdot n(c \ln n)$ rounds, the probability $\leq n^{-c}$.

Pf. Let $F[i,t]$ = event that process i fails to access database in rounds 1 through t.

By independence and previous claim, we have $\Pr[F[i,t]] \leq (1 - 1/(en))^t$.

Choose $t = \lceil e \cdot n \rceil$:

$$\Pr[F[i,t]] \leq (1 - 1/(en))^{\lceil e \cdot n \rceil} \leq (1 - 1/(en))^{en} \leq 1/e.$$

Choose $t = \lceil e \cdot n \rceil \lceil c \ln n \rceil$:

$$\Pr[F[i,t]] \leq (1/e)^{c \ln n} = n^{-c}. \ \square$$

Contention Resolution: Randomized Protocol

Claim. The probability that all processes succeed within $2e \cdot n \ln n$ rounds is $\geq 1 - 1/n$.

Pf. Let $F[t] =$ event that at least one of the n processes fails to access database in any of the rounds 1 through t.

Union bound. Given events $E_1, \cdots, E_n$,

$$\Pr[\cup_{i=1}^{n} E_i] \leq \sum_{i=1}^{n} \Pr[E_i].$$

$$\Pr[F[t]] = \Pr[\cup_{i=1}^{n} F[i,t]] \leq \sum_{i=1}^{n} \Pr[F[i,t]] \leq n(1 - \frac{1}{en})^t.$$

Choosing $t = 2\lceil en \rceil \lceil \ln n \rceil$ yields $Pr[F[t]] \leq n \cdot n^{-2} = 1/n$. $\square$

Expectation

- Expectation. Given a discrete random variables X, its expectation $E[X]$ is defined by:

$$E[X] = \sum_{j=0}^{\infty} j \Pr[X = j].$$

- *Waiting for a first success*. Coin is heads with probability p and tails with probability $1 - p$. How many independent flips X until first heads?

$$\begin{aligned} E[X] &= \sum_{j=0}^{\infty} j \Pr[X = j] = \sum_{j=0}^{\infty} j(1-p)^{j-1} p \\ &= p \sum_{j=0}^{\infty} j(1-p)^{j-1} = \frac{p}{p^2} = \frac{1}{p}. \square \end{aligned}$$

Expectation: Two Properties

- *Useful property*. If X is a $0/1$ random variable, $E[X] = \Pr[X = 1]$.

Pf. $E[X] = \sum_{j=0}^{\infty} j\Pr[X = j] = \sum_{j=0}^{1} j\Pr[X = j] = \Pr[X = 1].\square$

- *Linearity of expectation*. Given two random variables X and Y defined over the same probability space, $E[X + Y] = E[X] + E[Y]$.

- *Benefit*. Decouples a complex calculation into simpler pieces.

Guessing Cards

Game. Shuffle a deck of n cards; turn them over one at a time; try to guess each card.

- *Memoryless guessing*. Can't even remember what's been turned over already. Guess a card from full deck uniformly at random.

Claim. The expected number of correct guesses is 1.

Pf. Let $X_i = 1$ if i^{th} prediction is correct and 0 otherwise.

Let X = number of correct guesses $= X_1 + \cdots + X_n$.

$E[X_i] = \Pr[X_i = 1] = 1/n$.

$E[X] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/n = 1$. $\square$

Guessing Cards

- *Guessing with memory*. Guess a card uniformly at random from cards not yet seen.

Claim. The expected number of correct guesses is $\Theta(\log n)$.

Pf. Let $X_i = 1$ if i^{th} prediction is correct and 0 otherwise.

Let X = number of correct guesses $= X_1 + \cdots + X_n$.

$E[X_i] = \Pr[X_i = 1] = 1/(n - i + 1)$.

$E[X] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/2 + 1 = H(n)$. $\square$

Coupon Collector

- *Coupon collector.* Each box of cereal contains a coupon. There are n different types of coupons. Assuming all boxes are equally likely to contain each coupon, how many boxes before you have ≥ 1 coupon of each type?

Claim. The expected number of steps is $\Theta(n \log n)$.

Pf. Phase j = time between j and $j+1$ distinct coupons.

Let X_j = number of steps you spend in phase j.

Let X = number of steps in total $= X_0 + \cdots + X_{n-1}$.

$$E[X] = \sum_{j=0}^{n-1} E[X_j] = \sum_{j=0}^{n-1} \frac{n}{n-j} = n \sum_{i=1}^{n} \frac{1}{i} = nH(n). \quad \square$$

Maximum 3-Satisfiability

- *MAX-3SAT*. Given *3-SAT* formula, find a truth assignment that satisfies as many clauses as possible.

$$
\begin{aligned}
C_1 &= x_2 \vee \bar{x}_3 \vee \bar{x}_4 \\
C_2 &= x_2 \vee x_3 \vee \bar{x}_4 \\
C_3 &= \bar{x}_1 \vee x_2 \vee x_4 \\
C_4 &= \bar{x}_1 \vee x_2 \vee x_4 \\
C_5 &= x_1 \vee \bar{x}_2 \vee \bar{x}_4
\end{aligned}
$$

Remark. *NP*-hard search problem.

- *Simple idea*. Flip a coin, and set each variable true with probability $\frac{1}{2}$, independently for each variable.

Maximum 3-Satisfiability: Analysis

Claim. Given a *3-SAT* formula with k clauses, the expected number of clauses satisfied by a random assignment is $7k/8$.

Pf. Consider random variable

$$Z_j = \begin{cases} 1 & \text{if clause } C_j \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}$$

Let $Z =$ number of clauses satisfied by assignment.

$$\begin{aligned} E[Z] &= \sum_{j=1}^{k} E[Z_j] \\ &= \sum_{j=1}^{k} \Pr[\text{clause } C_j \text{ is satisfied}] \\ &= \frac{7}{8}k. \square \end{aligned}$$

Maximum 3-Satisfiability: Analysis

- *Corollary*. For any instance of *3-SAT*, there exists a truth assignment that satisfies at least a $7/8$ fraction of all clauses.

Pf. Random variable is at least its expectation some of the time. □

- Probabilistic method. [Paul Erdös] Prove the existence of a non-obvious property by showing that a random construction produces it with positive probability!

Maximum 3-Satisfiability: Analysis

Q. Can we turn this idea into a $7/8$-approximation algorithm?

A. Yes.

Lemma 1

The probability that a random assignment satisfies $\geq 7k/8$ *clauses is at least* $1/(8k)$.

Pf. Let p_j be probability that exactly j clauses are satisfied.

Let p be probability that $\geq 7k/8$ clauses are satisfied.

$$\begin{aligned}\frac{7}{8}k = E[Z] &= \sum_{j\geq 0} jp_j = \sum_{j<7k/8} jp_j + \sum_{j\geq 7k/8} jp_j \\ &\leq \frac{7k-1}{8}\sum_{j<7k/8} p_j + k\sum_{j\geq 7k/8} p_j \\ &\leq \frac{7k-1}{8}\times 1 + kp.\end{aligned}$$

Rearranging terms yields $p \geq 1/(8k)$. □

Maximum 3-Satisfiability: Analysis

- *Johnson's algorithm*. Repeatedly generate random truth assignments untilone of them satisfies $\geq 7k/8$ clauses.

Theorem 2

Johnson's algorithm is a 7/8-approximation algorithm.

By previous lemma, each iteration succeeds with probability $\geq 1/(8k)$.

The expected number of trials to find the satisfying assignment is at most $8k$. $\square$

Theorem 3 (Håstad 1997)

Unless P = NP, no ρ-approximation algorithm for MAX-3-SAT (and hence MAX-SAT) for any $\rho > 7/8$.

Homework

- Read Chapter 13 of the textbook.
- Exercise 1 & 3 in Chapter 13.