上海科技大学
ShanghaiTech University

CS240 Algorithm Design and Analysis

Lecture 20

Randomized Algorithms (Cont.)

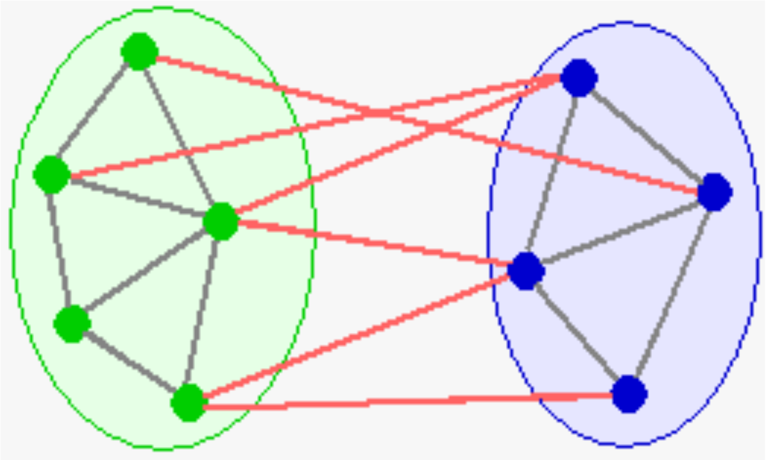
Quan Li

Fall 2023

2023.12.19

Max Cut

- We studied the Min-Cut problem, which is closely related to finding the max flow in a network.
- Max-Cut is the opposite of Min-Cut.
- Given a graph G, split vertices into two sides to maximize the number of edges between the sides.



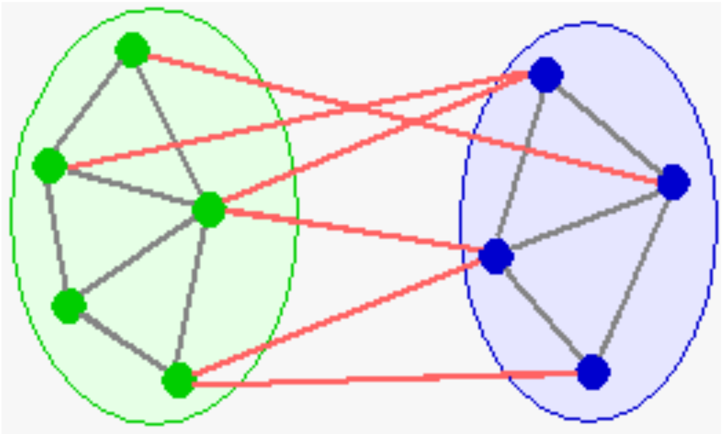
Max Cut

- Unlike Min-Cut, Max-Cut is NP-complete.
- We'll give a very simple randomized Monte Carlo 2-approximation algorithm.
  - Monte Carlo means the algorithm sometimes returns the wrong answer, i.e., a cut that's not a 2-approximation.
  - Monte Carlo also means the algorithm always runs in a fixed amount of time.

    - Put each node in a random side with probability ½.

Correctness

- Lemma In a graph with e edges, the algorithm produces a cut with expected size e/2.
- Proof Let X be a random variable equal to the size of the cut. We want to bound E[X].
  - For each edge e, let X_e be the indicator variable of whether e is in the cut.
    - i.e., X_e=1 if e is in the cut and 0 otherwise.
  - X= $\sum_e X_e$.
  - Given an edge e=(i,j), e is in the cut if i and j are on different sides.
  - Pr[e in cut]=Pr[(i in L) ∧ (j in R)] + Pr[(j in L) ∧ (i in R)]=1/4+1/4=1/2.
  - E[X_e]=1/2.
  - E[X]=e/2 by linearity of expectations.



立志成才报国裕民

Correctness

- Since a cut can have at most e edges, the e/2 edges the algorithm outputs in expectation is a 2 approximation.
- Note that we only bounded expected size of the algorithm's cut.
  - In any particular execution, the algorithm can output a cut that's smaller or larger than e/2.
    - On average, the cut has size e/2.

Contention Resolution

An example in distributed computing where randomization is necessary

Distributed Computing

- Distributed system
  - Set of autonomous nodes, working independently of each other
  - Nodes may be able to communicate, at a cost
  - Ex: Internet, computer cluster, sensor network
- Nodes need to coordinate to solve some problem
- Coordination can be done using communication. But communication is expensive
- By making nodes randomized, they can coordinate with minimal communication
- Randomization also simplifies symmetry breaking between nodes

Contention Resolution in a Distributed System

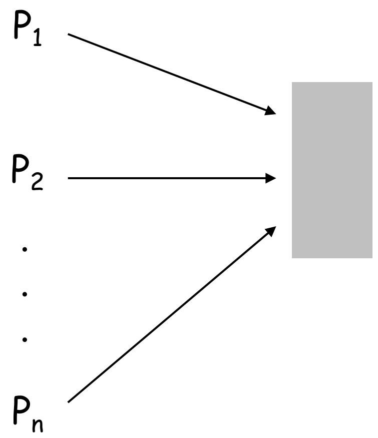
Contention resolution. Given n processes $P_1, \ldots, P_n$, each competing for access to a shared channel. If two or more processes access the channel simultaneously, all processes are locked out. Devise protocol to ensure all processes get through on a regular basis.

Assumption. Time is divided into rounds.

Restriction. Processes can't communicate, and they don't have id's.

Challenge. Need symmetry-breaking paradigm.

No deterministic protocol can solve the problem.



Contention Resolution: Randomized Protocol

Protocol. Each process requests access to the channel 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 $1/(e \cdot n) \le \Pr[S(i, t)] \le 1/(2n)$.

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

process i requests access

none of remaining n-1 processes request access

- Setting $p = 1/n$, we have $\Pr[S(i, t)] = 1/n \underbrace{(1 - 1/n)^{n-1}}_{\text{between } 1/e \text{ and } 1/2}$. ▪

Useful facts from calculus.

- $1/4 < (1 - 1/n)^n < 1/e < (1 - 1/n)^{n-1} < 1/2$
- $9/4 < (1 + 1/n)^n < e < (1 + 1/n)^{n+1} < 27/8$

Contention Resolution: Randomized Protocol

Claim. The probability that process i fails to access the channel in e·n rounds is at most 1/e. After e·n·c ln n rounds, the probability is at most 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 = e n: $$\Pr[F(i,t)] \leq \left(1-\tfrac{1}{en}\right)^{en} \leq \tfrac{1}{e}$$
- Choose t = (e n) (c ln n): $$\Pr[F(i,t)] \leq \left(\tfrac{1}{e}\right)^{c\ln n} = n^{-c}$$

Contention Resolution: Randomized Protocol

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

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.

$$\Pr[F[t]] = \Pr\left[\bigcup_{i=1}^{n} F[i,t]\right] \le \sum_{i=1}^{n} \Pr[F[i,t]] \le n \cdot n^{-c}$$

union bound previous slide

- Choosing $t = 2\, e\, n \ln n$ yields $\Pr[F[t]] \le n \cdot n^{-2} = 1/n$. ▪

Union bound. Given events $E_1, \ldots, E_n$, independent or not,

$$\Pr\left[\bigcup_{i=1}^{n} E_i\right] \le \sum_{i=1}^{n} \Pr[E_i]$$

Global Minimum Cut

A problem for which the best-known randomized algorithm is faster than the best-known deterministic algorithm

Global Minimum Cut

Global min cut. Given a connected, undirected graph G = (V, E), find a cut (A, B) of minimum cardinality.

Applications. Partitioning items in a database, identify clusters of related documents, network reliability, network design, circuit design, TSP solvers.

Network flow solution.

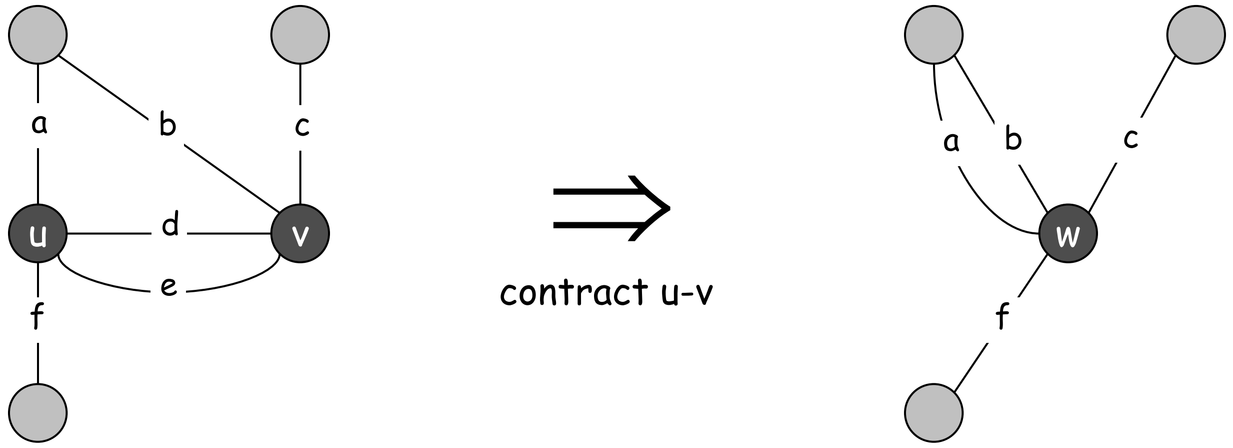
- Replace every edge (u, v) with two antiparallel edges (u, v) and (v, u).
- Pick some vertex s and compute min s-v cut separating s from each other vertex $v \in V$.

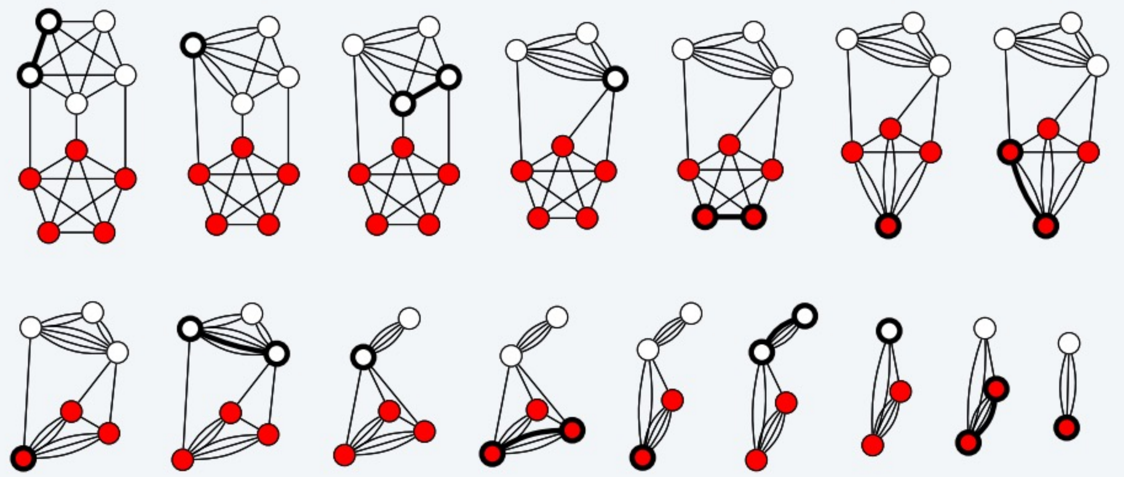
False intuition. Global min-cut is harder than min s-t cut.

Contraction Algorithm

Contraction algorithm. [Karger 1995]

- Pick an edge $e = (u, v)$ uniformly at random.
- Contract edge e.
  - replace u and v by a single new supernode w
  - preserve edges, updating endpoints of u and v to w
  - keep parallel edges, but delete self-loops
- Repeat until graph has just two supernodes v_1 and v_2.
- Return the cut (between the two supernodes).





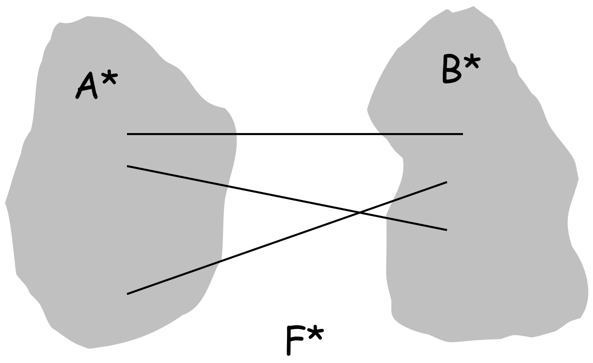
Reference: Thore Husfeldt

Contraction Algorithm

Claim. The contraction algorithm returns a min cut with prob $\geq 2/n^2$.

Pf. Consider a global min-cut (A^*, B^*) of G. Let F^* be edges with one endpoint in A^* and the other in B^*. Let $k = |F^*|$ = size of min cut.

- In first step, algorithm contracts an edge in F^* with prob $k / |E|$.
- Every node has degree $\geq k$ since otherwise (A^*, B^*) would not be min-cut. $\Rightarrow$ $|E| \geq \frac{1}{2}kn$.
- Thus, algorithm contracts an edge in F^* with probability $\leq 2/n$.



Contraction Algorithm

Claim. The contraction algorithm returns a min cut with prob $\geq 2/n^2$.

Pf. Consider a global min-cut (A*, B*) of G. Let F* be edges with one endpoint in A* and the other in B*. Let k = |F*| = size of min cut.

- Let G' be graph after j iterations. There are n' = n-j supernodes.
- Suppose no edge in F* has been contracted. The min-cut in G' is still k.
- Since value of min-cut is k, $|E'| \geq \frac{1}{2}kn'$ -> $k/|E'| <= 2/n'$
- Thus, algorithm contracts an edge in F* with probability $\leq 2/n'$.

- Let E_j = event that an edge in F* is not contracted in iteration j.

$$\begin{aligned}
\Pr[E_1 \cap E_2 \cdots \cap E_{n-2}] &= \Pr[E_1] \times \Pr[E_2 \mid E_1] \times \cdots \times \Pr[E_{n-2} \mid E_1 \cap E_2 \cdots \cap E_{n-3}] \\
&\geq \left(1 - \tfrac{2}{n}\right)\left(1 - \tfrac{2}{n-1}\right) \cdots \left(1 - \tfrac{2}{4}\right)\left(1 - \tfrac{2}{3}\right) \\
&= \left(\tfrac{n-2}{n}\right)\left(\tfrac{n-3}{n-1}\right) \cdots \left(\tfrac{2}{4}\right)\left(\tfrac{1}{3}\right) \\
&= \tfrac{2}{n(n-1)} \\
&\geq \tfrac{2}{n^2}
\end{aligned}$$

Contraction Algorithm

Amplification. To amplify the probability of success, run the contraction algorithm many times.

Claim. If we repeat the contraction algorithm n^2 times with independent random choices and return the best cut found, then the algorithm finds the min-cut with constant probability.

Pf. By independence, the probability of failure is at most

$$\left(1-\frac{2}{n^2}\right)^{n^2} = \left[\left(1-\frac{2}{n^2}\right)^{\frac{1}{2}n^2}\right]^2 \le \left(e^{-1}\right)^2 = \frac{1}{e^2}$$

Global Min Cut: Context

Remark. Overall running time is slow since we perform $O(V^2)$ iterations, and each takes $O(E \log V)$ time (we always merge the vertex with smaller degree into the other).

Best known. [Karger 2000] $O(E \log^3 V)$.

faster than best known deterministic global min cut algorithm

Random Variables and Expectations

A Quick Review of Probability Theory

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

$$E[X] = \sum i \cdot \Pr[X = i]$$

Q: Roll a 6-sided dice. What is the expected value?

A: ?

Q: Roll two dice. What is the expected maximum value?

A: ?

Expectation: Two Properties

Indicator random variables. If X only takes 0 or 1, $E[X] = \Pr[X = 1]$.

Linearity of expectation. Given two random variables X and Y (not necessarily independent),

$$E[X + Y] = E[X] + E[Y].$$

Remark: $E[XY] = E[X]E[Y]$ only when X and Y are independent.

Example. Shuffle a deck of n cards; turn them over one at a time; try to guess each card. Suppose you can't remember what's been turned over already, and just guess a card from full deck uniformly at random.

Q. What's the expected number of correct guesses?

A. (surprisingly effortless using linearity of expectation)

- Let $X_i = 1$ if i^{th} guess is correct and 0 otherwise.
- Let $X =$ number of correct guesses $= X_1 + \cdots + X_n$.
- $E[X_i] = \Pr[X_i = 1] = 1/n$.
- $E[X] = E[X_1] + \cdots + E[X_n] = 1/n + \cdots + 1/n = 1$.

Guessing Cards with Memory

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

Q. What's the expected number of correct guesses?

A.

- Let $X_i = 1$ if i^{th} guess is correct and 0 otherwise.
- Let X = number of correct guesses $= X_1 + \cdots + X_n$.
- $E[X_i] = \Pr[X_i = 1] = 1/(n - i + 1)$.
- $E[X] = E[X_1] + \cdots + E[X_n] = \frac{1}{n} + \cdots + \frac{1}{2} + \frac{1}{1} = \Theta(\log n)$.

立志成才 报国裕民

The Birthday Paradox

Problem: Suppose there are $n = 365$ days in a year, and in a room of k people, each person's birthday falls in any one of the n days with equal probability. How large should k be for us to expect two people with the same birthday?

Analysis:

- Define $X_{ij} = 1$ if person i and person j have the same birthday, and 0 otherwise.
- We know $E\left[X_{ij}\right] = \Pr\left[X_{ij} = 1\right] = 1/n$.
- Let $X = \sum_{1\le i<j\le k} X_{ij}$ be the number of pairs of people having the same birthday.
- We have
$$E[X] = E\left[\sum_{1\le i<j\le k} X_{ij}\right] = \binom{k}{2}\frac{1}{n} = \frac{k(k-1)}{2n}$$
- So, when $\frac{k(k-1)}{2n} \ge \frac{(k-1)^2}{2n} \ge 1$, or $k \ge \sqrt{2n} + 1 \approx 28$, we expect to see at least one pair of people having the same birthday.

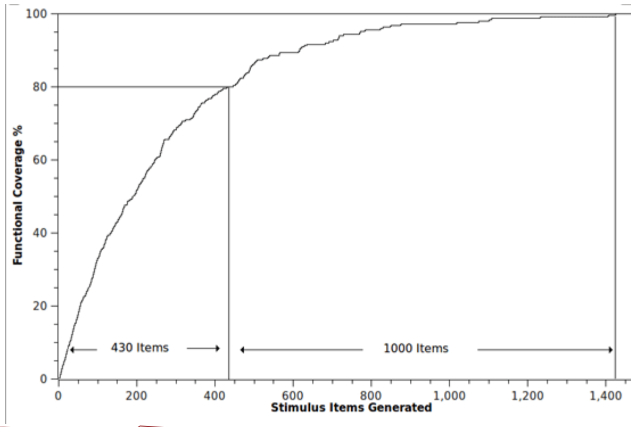
Coupon Collector

Coupon collector. Each box of cereal contains a coupon. There are n different types of coupons. Assuming a box contains each type of coupon equally likely, how many boxes do you need to open to have at least one coupon of each type?

Solution.

- Stage i = time between i and $i+1$ distinct coupons.
- Let X_i = number of steps you spend in stage i.
- Let X = number of steps in total $= X_0 + X_1 + \cdots + X_{n-1}$.

$$E[X] = \sum_{i=0}^{n-1} E[X_i] = \sum_{i=0}^{n-1} \frac{n}{n-i} = n \sum_{i=1}^{n} \frac{1}{i} = \Theta(n \log n)$$



prob of success $= (n-i)/n$

$\Rightarrow$ expected waiting time $= n/(n-i)$

MAX 3-SAT

An extremely simple randomized approximation algorithm

Maximum 3-Satisfiability

exactly 3 distinct literals per clause

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

$$
\begin{aligned}
C_1 &= x_2 \vee \overline{x_3} \vee \overline{x_4} \\
C_2 &= x_2 \vee x_3 \vee \overline{x_4} \\
C_3 &= \overline{x_1} \vee x_2 \vee x_4 \\
C_4 &= \overline{x_1} \vee \overline{x_2} \vee x_3 \\
C_5 &= x_1 \vee \overline{x_2} \vee \overline{x_4}
\end{aligned}
$$

Remark. NP-hard search problem.

Simple idea. Flip a coin, and set each variable true with probability ½, 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 = total number of clauses satisfied.

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

linearity of expectation

Maximum 3-Satisfiability: Analysis

Lemma. The probability that a random assignment satisfies ≥ 7k/8 clauses is at least 1/(8k).

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

We start by writing

$$\frac{7}{8}k = \sum_{j=0}^{k} jp_j = \sum_{j<7k/8} jp_j + \sum_{j\geq 7k/8} jp_j$$
$$\leq \sum_{j<7k/8} k'p_j + \sum_{j\geq 7k/8} kp_j$$
$$= k'(1-p) + kp \leq k' + kp$$

Hence, $kp \geq \frac{7}{8}k - k'$

But $\frac{7}{8}k - k' \geq 1/8$ (k′ is the largest natural number that is strictly smaller than $\frac{7}{8}$k)

So

$$p \geq \frac{\frac{7}{8}k - k'}{k} \geq \frac{1}{8k}.$$

Next Time:
Randomized algorithms (Cont.)