Lecture 15

Thursday, July 13, 2017 12:50 PM

Randomization

Algorithmic design patterns.

- Greedy.
- Divide-and-conquer.
- Dynamic programming.
- Network flow.
- Randomization.

in practice, access to a pseudo-random number generator

Randomization. Allow fair coin flip in unit time.

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

Ex. Symmetry breaking protocols, graph algorithms, quicksort, hashing, load balancing, Monte Carlo integration, cryptography.

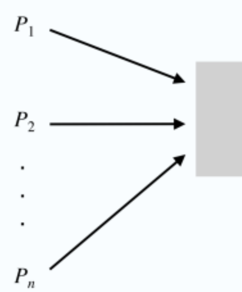
2

Contention resolution in a distributed system

Contention resolution. Given n processes $P_1, \ldots, P_n$, 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.



4

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

process i requests access — none of remaining n-1 processes request access

$p = \frac{1}{n}$ $\quad \frac{1}{e} \leq \left(1 - \frac{1}{n}\right)^{n-1} \leq \frac{1}{2}$

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

value that maximizes Pr[S(i, t)] — between 1/e and 1/2

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 $1/e$.

5

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 $\le n^{-c}$.

$\Pr[F[i,t] \text{ in one round}] = 1 - \Pr[\text{success}] \le \left(1 - \frac{1}{en}\right)$

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

$\Pr[F[i,t]] \le (1 - 1/(en))^t$.

- Choose $t = \lceil e \cdot n \rceil$: $\quad \Pr[F(i,t)] \le \left(1 - \frac{1}{en}\right)^{\lceil en \rceil} \le \left(1 - \frac{1}{en}\right)^{en} \le \frac{1}{e}$
- Choose $t = \lceil e \cdot n \rceil \lceil c \ln n \rceil$: $\quad \Pr[F(i,t)] \le \left(\frac{1}{e}\right)^{c \ln n} = n^{-c}$

$\underbrace{en \quad en \quad \ldots \quad en}_{c \ln n}$

$\forall m,\ \left(1 - \frac{1}{m}\right)^m \to \frac{1}{e}$

6

Contention Resolution: randomized protocol

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

$c = 2$

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.

$$\Pr[F[t]] = \Pr\left[\bigcup_{i=1}^{n} F[i,t]\right] \le \sum_{i=1}^{n} \Pr[F[i,t]] \le n\cancel{\left(1 - \frac{1}{en}\right)^t}$$

union bound ↑ previous slide ↑

$\le \frac{1}{n^2}$

$n \cdot \frac{1}{n^2} = \frac{1}{n}$

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

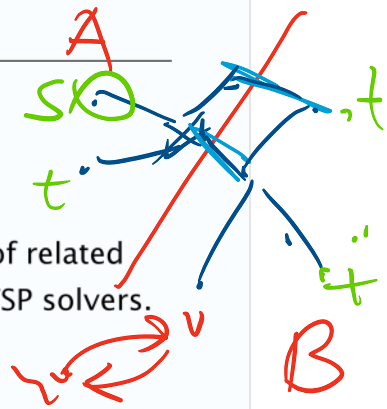
$\Pr[\text{overall success}] \ge 1 - \frac{1}{n}$

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

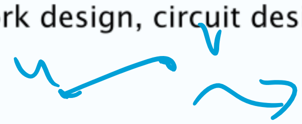
7

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 $v \in V$.

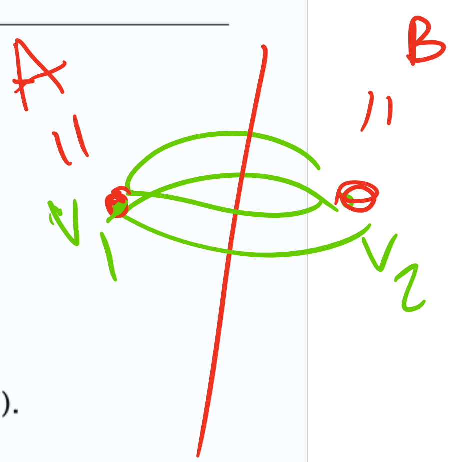
False intuition. Global min-cut is harder than min s-t cut.

9

Contraction algorithm

Contraction algorithm. [Karger 1995]

- Pick an edge $e = (u, v)$ uniformly at random.
- Contract edge e.
  - replace u and v by single new super-node w
  - preserve edges, updating endpoints of u and v to w
  - keep parallel edges, but delete self-loops
- Repeat until graph has just two nodes v_1 and v_2.
- Return the cut (all nodes that were contracted to form v_1).



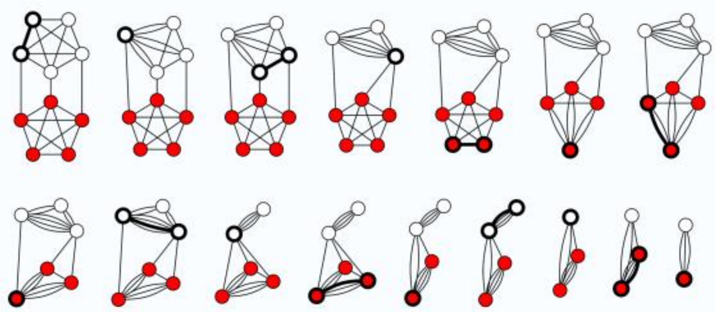
$w = \{u, v\}$

10

Contraction algorithm

Contraction algorithm. [Karger 1995]

- Pick an edge $e = (u, v)$ uniformly at random.
- Contract edge e.
  - replace u and v by single new super-node w
  - preserve edges, updating endpoints of u and v to w
  - keep parallel edges, but delete self-loops
- Repeat until graph has just two nodes v_1 and v_2.
- Return the cut (all nodes that were contracted to form v_1).



Reference: Thore Husfeldt

11

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

$2 \cdot |E| = \sum_v \deg(v) \geq n \cdot k$

$\boxed{\text{each } \deg(v) \geq k}$

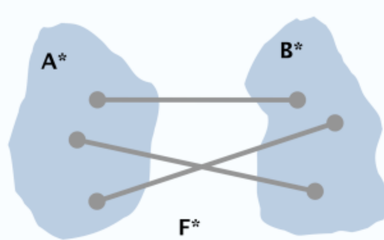
$\Pr[\text{pick edge in } F^*] = \frac{|F^*|}{|E|} \leq \frac{\cancel{k}}{\frac{1}{2} \cdot \cancel{k} n} = \frac{2}{n}$

$k = 3$

$<k$

A

B



12

$= \frac{2}{n}$

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 $\underbrace{n' = n - j}$ supernodes.
- Suppose no edge in F^* has been contracted. The min-cut in G' is still k.
- Since value of min-cut is k, $|E'| \geq \frac{1}{2} k n'$.
- 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.

$\Pr[\text{pick edge in } F^*] = \frac{|F^*|}{|E'|} \leq \frac{K}{\frac{1}{2} k \cdot n'} = \frac{2}{n'}$

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

$\cancel{\left(\frac{n-4}{n-2}\right)}$

$\Pr[A \,\&\, B] = \Pr[A] \cdot \Pr[B \mid A]$

Contraction algorithm

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

with independent random choices,

Claim. If we repeat the contraction algorithm $n^2 \ln n$ times, then the probability of failing to find the global min-cut is $\leq 1 / n^2$.

repeat t times

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

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

$\boxed{(1 - 1/x)^x \leq 1/e}$

$\Pr[\text{fail}] \leq \left(1 - \frac{2}{n^2}\right)^t$

$\Pr[\text{success}] \geq 1 - \frac{1}{n^2}$

pick $t \sim n^2 \ln n$

14

Contraction algorithm: example execution

trial 1

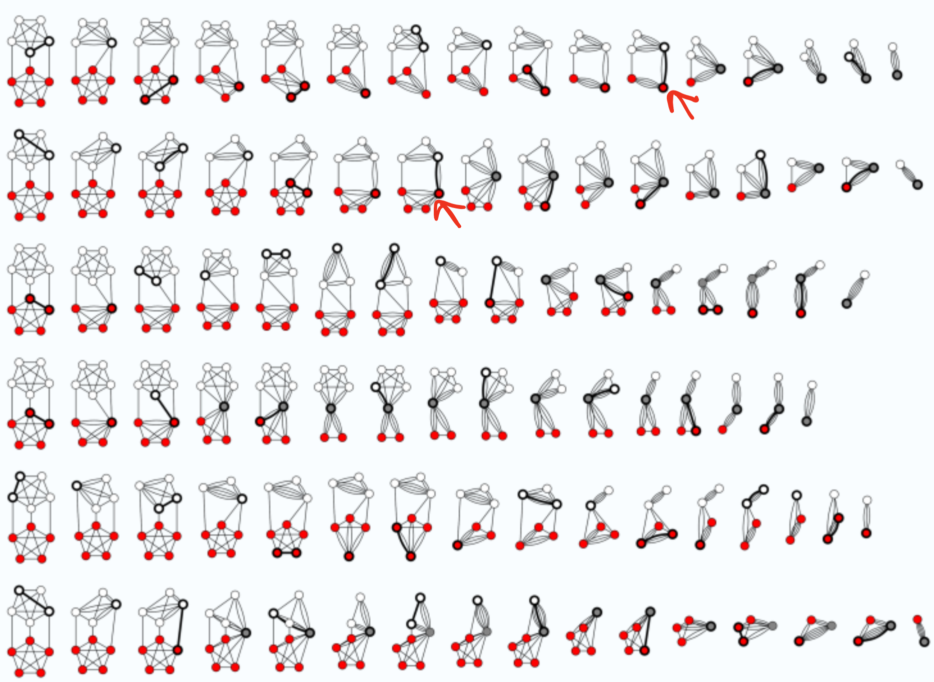
trial 2

trial 3

trial 4

trial 5
(finds min cut)

trial 6



...

Reference: Thore Husfeldt

15

Global min cut: context

Remark. Overall running time is slow since we perform $\Theta(n^2 \log n)$ iterations and each takes $\Omega(m)$ time.

Improvement. [Karger–Stein 1996] $O(n^2 \log^3 n)$.

- Early iterations are less risky than later ones: probability of contracting an edge in min cut hits 50% when $n / \sqrt{2}$ nodes remain.
- Run contraction algorithm until $n / \sqrt{2}$ nodes remain.
- Run contraction algorithm twice on resulting graph and return best of two cuts.

Extensions. Naturally generalizes to handle positive weights.

Best known. [Karger 2000] $O(m \log^3 n)$.

faster than best known max flow algorithm or deterministic global min cut algorithm

16