COMP 251

Algorithms & Data Structures (Winter 2022)

Extras – Randomization and Probabilistic Analysis

School of Computer Science
McGill University

Slides of Langer (2014) & Cormen et al., 2009 & Comp251-Fall McGill & Kleinberg & *Tardos*, 2006 & Lin & Devi (UNC)

Outline

- Extras.
  - Amortized Analysis.
  - Randomized algorithms.
  - Probabilistic Analysis.
  - Review Final Exam.

Randomization

Principle: Allow flip a fair coin in unit time.

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

by applying a random rule

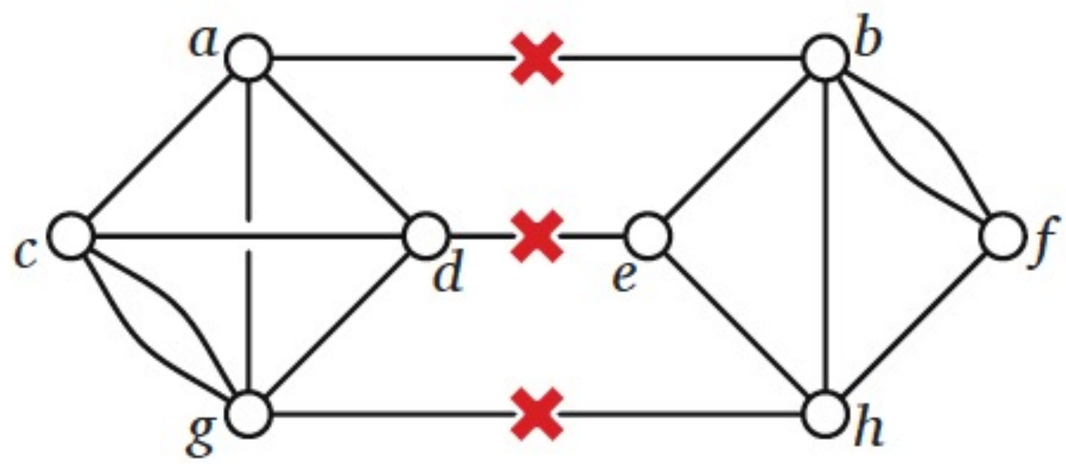
Examples:

- Quicksort
- Graph Algorithms
- Hashing
- Monte-Carlo integration
- Distributed systems
- Cryptography

Global Min Cut

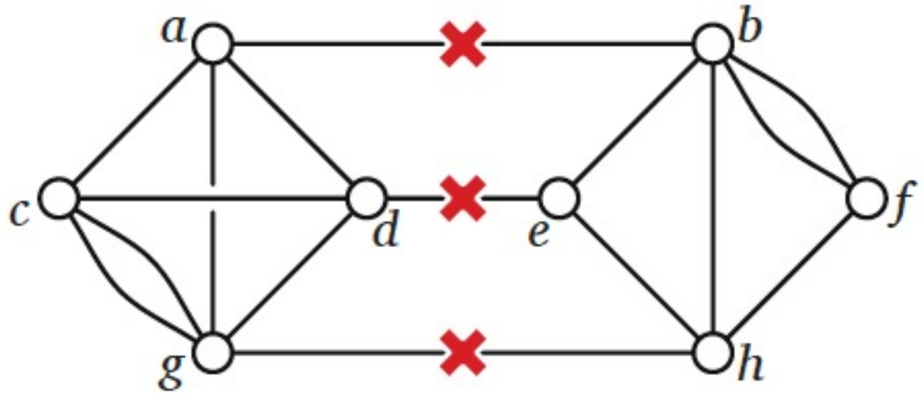
Definition: Given a connected, undirected graph G=(V,E), find a cut with minimum cardinality.

- A cut partitions the nodes of G into two nonempty subsets.
- The size of the cut is the number of crossing edges, which have one endpoint in each subset.
- A minimum cut in G is a cut with the smallest number of crossing edges
- The same graph may have several minimum cuts.



Global Min Cut

Definition: Given a connected, undirected graph G=(V,E), find a cut with minimum cardinality.



not the best way

undirected

Network solution:

- Replace every edge (u,v) with 2 antiparallel edges (u,v) & (v,u)
- Pick some vertex s, and compute min s-v cut for each other vertex v.
  - This is $n - 1$ directed minimum-cut computations
- Fastest deterministic algorithm run in $O(n^3)$ and it is complex.

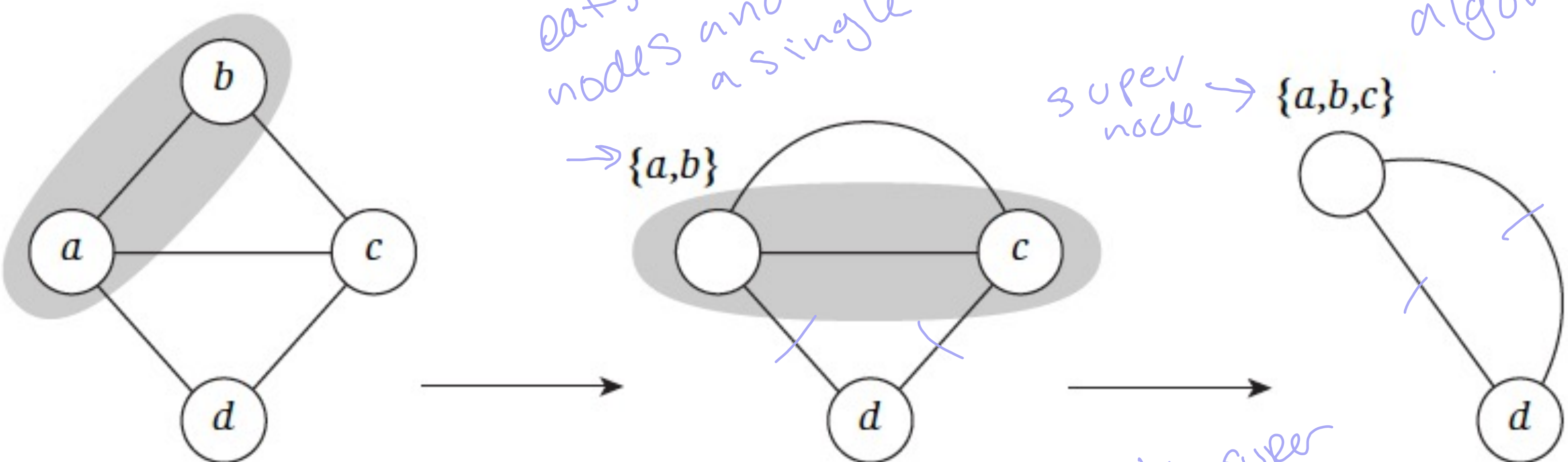
False Intuition: Global min-cut is harder that min s-t cut!

Contraction Algorithm

- The Contraction Algorithm works with a connected multigraph.
  - This is an undirected graph that is allowed to have multiple “parallel” edges between the same pair of nodes.
- Uses a primitive operation called edge contraction.
  - Requires O(n) time

this is not the fastest algorithm

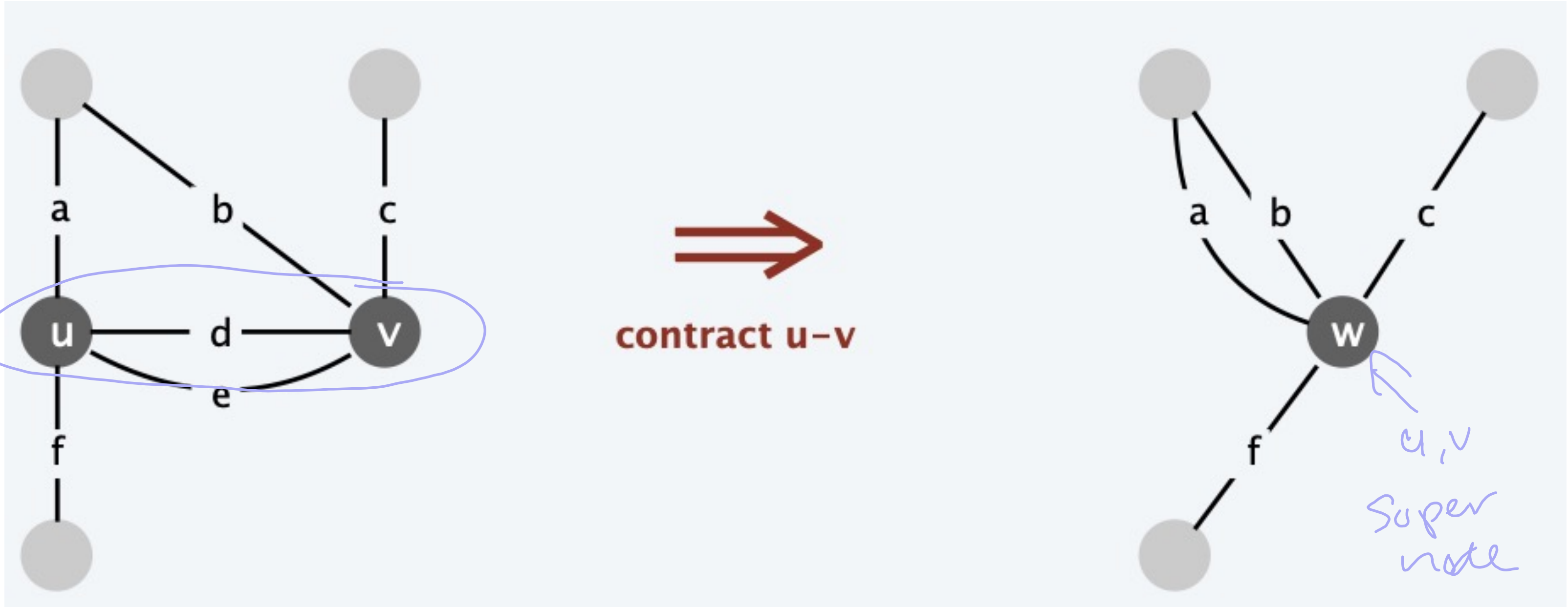
eats two nodes and creates a single node



create super node and delete self edges

Contraction Algorithm

- The Contraction Algorithm works with a connected multigraph.
  - This is an undirected graph that is allowed to have multiple “parallel” edges between the same pair of nodes.
- Uses a primitive operation called edge contraction.
  - Requires O(n) time

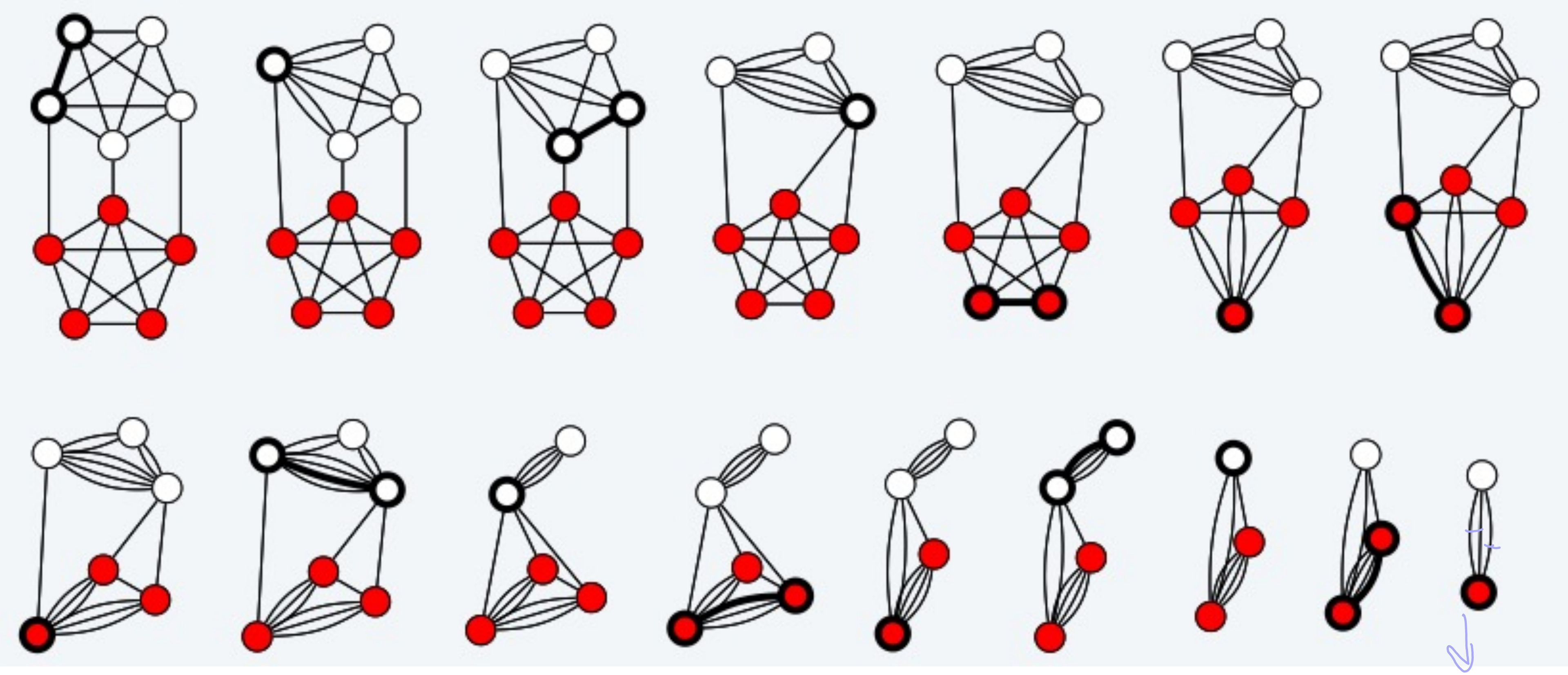


Contraction Algorithm

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_1.
- Return the cut (all nodes that were contracted to form v_1).

Contraction Algorithm



3

Contraction Algorithm

Contraction(V,E):

While $|V| > 2$ do

Choose $e \in E$ uniformly at random

$G \leftarrow G - \{e\}$ // contract G

return { the only cut in G }

Randomization

- The algorithm is making random choices,
  - There is some probability that it will succeed in finding a global min-cut and some probability that it won't.
  - There are exponentially many possible cuts of G.
    - One might imagine that the probability of success is exponentially small.
      - what's favoring the minimum cut in the process?

Contraction Algorithm

Contraction(V,E):

O(n^2)

While $|V| > 2$ do

Choose $e \in E$ uniformly at random

Randomization

$G \leftarrow G - \{e\}$ // contract G

return { the only cut in G }

- The algorithm is making random choices,
  - There is some probability that it will succeed in finding a global min-cut and some probability that it won't.
  - There are exponentially many possible cuts of G.
    - One might imagine that the probability of success is exponentially small.
      - what's favoring the minimum cut in the process?

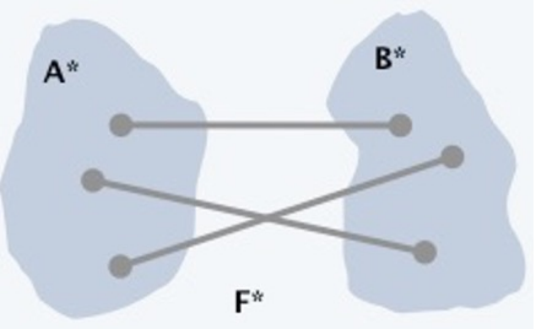
Contraction Algorithm

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

not on exam

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 $\leq 2 / n$.



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, $|E'| \geq \frac{1}{2} k n'$.
- 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.

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

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 $\boxed{n^2 \ln n}$ times, then the probability of failing to find the global min-cut is $\leq 1 / n^2$.

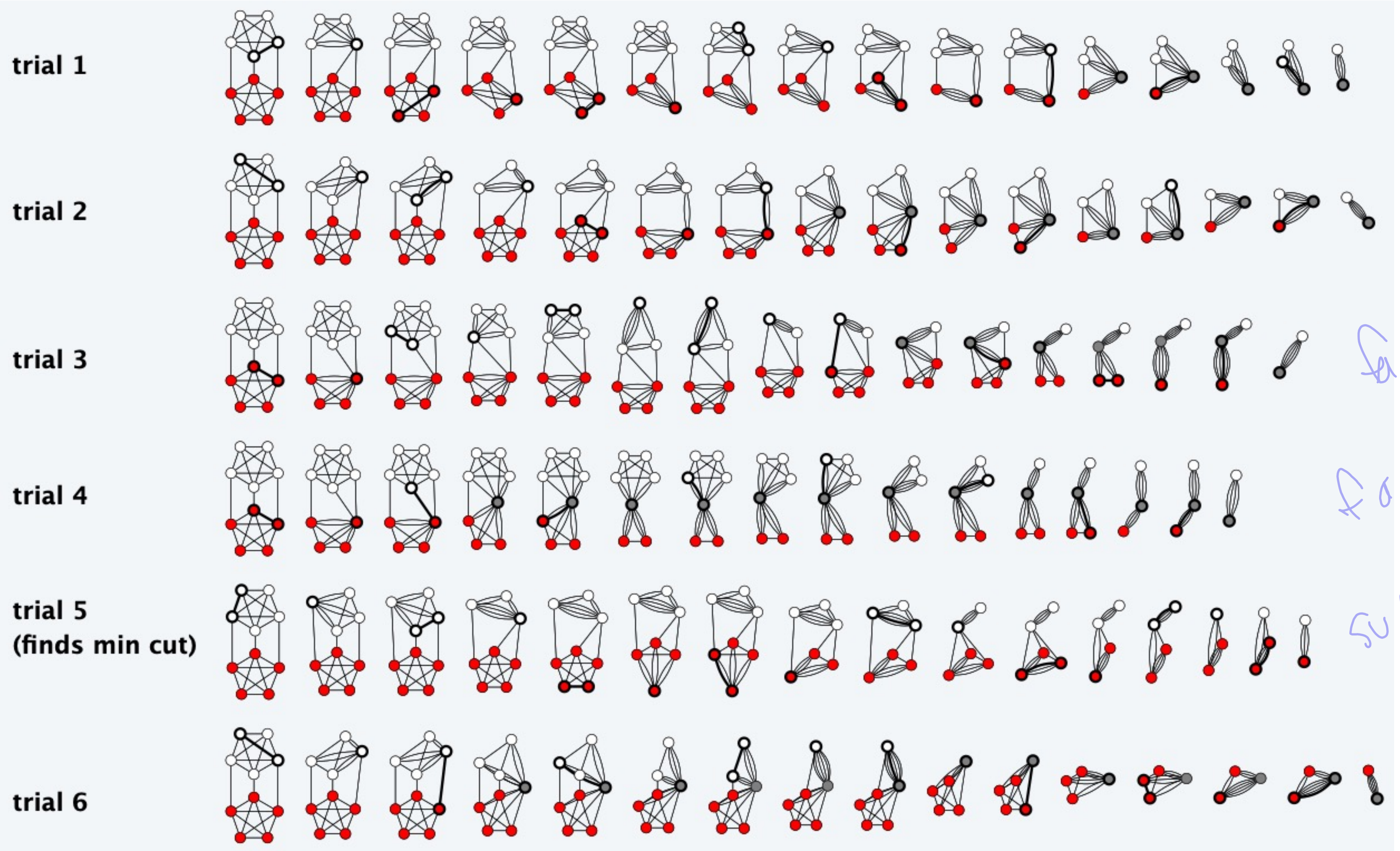
Pf. By independence, the probability of failure is at most

prob of finding right answer

$$\underbrace{\left(1 - \frac{2}{n^2}\right)}_{\text{prob of failure}}{}^{n^2 \ln n} = \left[\left(1 - \frac{2}{n^2}\right)^{\frac{1}{2}n^2}\right]^{2 \ln n} \leq \left(e^{-1}\right)^{2 \ln n} = \frac{1}{n^2}$$

$(1 - 1/x)^x \leq 1/e$

Contraction Algorithm



fail
fail
fail
fail
success
fail

Contraction Algorithm

this was the basic idea

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

- We can increase the number of iterations, but it is usually overkill.
  - We're facing a tradeoff between the speed of the algorithm and its probability of success.

Improvement: variations of contraction alg (faster

- As the graph shrinks, the probability of contracting an edge in the minimum cut increases. In other words, early iterations are less risky than later ones.
  - At first the probability is quite small, only 2/n, but near the end of execution, when the graph has only three vertices, we have a 2/3 chance of screwing up!
- To group the first several random contractions a "safe" phase, so that the cumulative probability of screwing up is relatively small and a "dangerous" phase, which is much more likely to screw up.
  - To get around the danger of the dangerous phase, we use amplification: we run the dangerous phase four times and keep the best of the four answers.
- $O(n^2 \log^3 n)$
- Best known $O(m \log^3 n)$

Outline

- Extras.
  - Amortized Analysis.
  - Randomized algorithms.
  - Probabilistic Analysis.
  - Review Final Exam.


time for questions