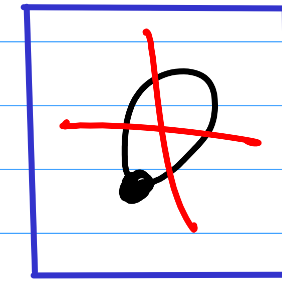
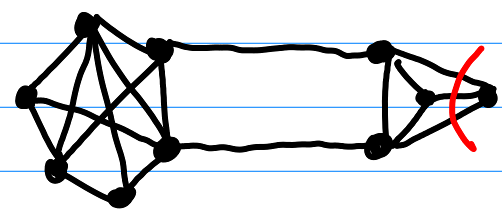
Combinatorial Optimization: Min-Cut

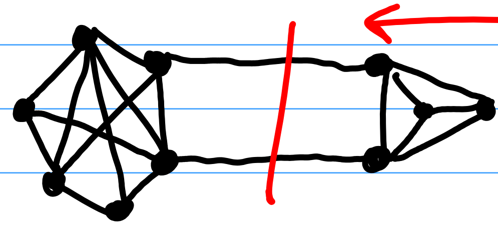


- $G=(V,E)$ is an undirected graph with no self loops
- A cut of G is a set of edges whose removal disconnects G.



← cut of size 3

- A min-cut is a cut of minimum cardinality



← cut of size 2 (min cut)

Min-Cut Background

Best deterministic MinCut algorithms run in $O(n^3)$ time

- complicated
- difficult to implement
- difficult to understand

We want something

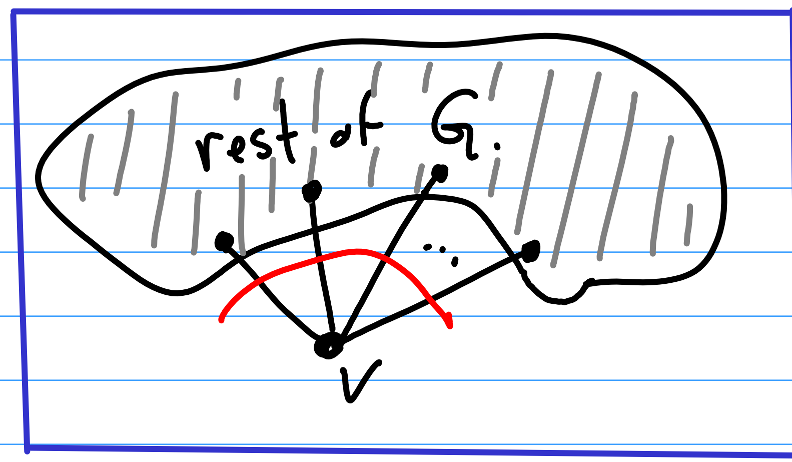
- simple
- easy to implement
- easy to understand (the algorithm, not necessarily the analysis)

How Big Can a MinCut Be?

Let C be any min-cut of a graph G with n vertices and m edges. Then

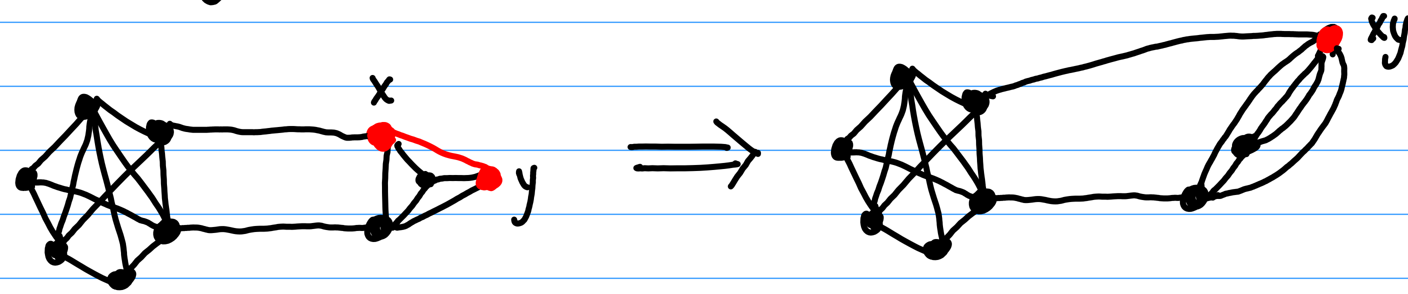
$$|C| \leq \frac{2m}{n}$$

Proof: G has a vertex v whose degree is at most $2m/n$. v's incident edges form a cut of size $\leq 2m/n$.



Edge Contraction

Contracting G at edge (x,y) involves identifying x and y, and removing any self loops to get a new graph $G/(x,y)$



Notice: The min-cut of G is no bigger than the min-cut of $G/(x,y)$

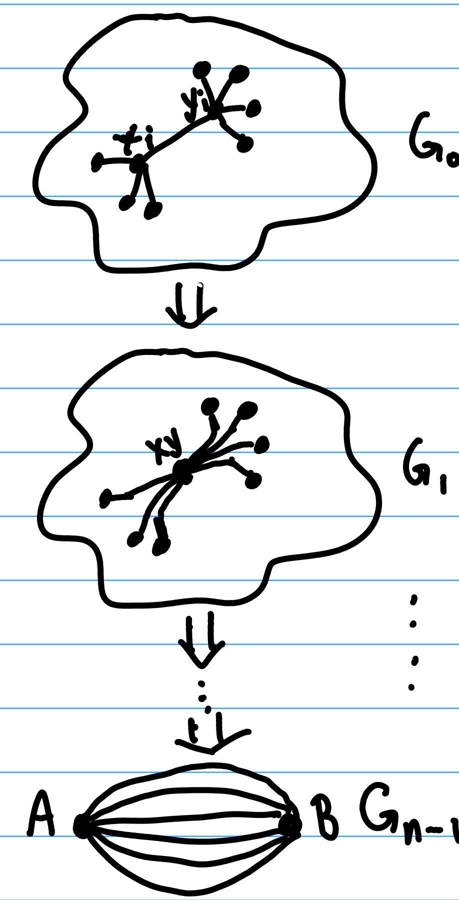
Notice: If C is a min-cut in G and $(x,y) \notin C$ then C is also a min-cut in $G/(x,y)$

Notice: If implemented carefully, edge contraction takes $O(n)$ time

Karger's Contraction Algorithm

$G_i = (V_i, E_i)$
$n_i = |V_i|$

```
KargerMinCut(G)
    G_0 ← G
    for i=1 to |V|-2 do
        select a random edge (x_i, y_i) in E(G_{i-1})
        G_i ← G_{i-1} / (x_i, y_i)
    return all edges in G_{n-1}
```



- KargerMinCut runs in $O(n^2)$ time
- Does it correctly return a min-cut of G?

Correctness of KargerMinCut

Let C be some min-cut in G

$$\Pr\{(x_1, y_1) \in C\} = \frac{|C|}{|E|} = \frac{|C|}{|E_0|} \leq \frac{2|E_0|}{|E_0| \cdot |V_0|} = \frac{2}{n}$$

$\therefore \Pr\{C \text{ survives at time } 1.\} \geq 1 - \frac{2}{n}$

$$\Pr\{C \text{ survives at time } i \mid C \text{ survives at time } i-1\} \geq 1 - \frac{2}{n-i+1}$$

$\therefore \Pr\{C \text{ survives at time } n-2\}$

$$\geq \left(1 - \frac{2}{n}\right) \cdot \left(1 - \frac{2}{n-1}\right) \cdot \left(1 - \frac{2}{n-2}\right) \cdot \cdots \cdot \left(1 - \frac{2}{3}\right)$$

$$= \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\left(\frac{n-4}{n-2}\right) \cdot \cdots \cdot \left(\frac{1}{3}\right) = \frac{2(n-2)!}{n!} = \frac{2}{n(n-1)} \geq \frac{1}{n^2}$$

$\therefore \Pr\{\text{KargerMinCut is correct}\} \geq \frac{1}{n^2}$

$$\boxed{\left(1-\frac{1}{k}\right)^k \le \frac{1}{e}}$$

Improving KargerMinCut

Theorem: Algorithm KargerMinCut runs in $O(n^2)$ time and correctly reports a min-cut with probability at least $1/n^2$

KargerMinCut*:
Run KargerMinC $cn^2 \ln n$ times and report the best solution found

$$\Pr\{\text{KargerMinCut* is incorrect}\} \le \left(1-\frac{1}{n^2}\right)^{cn^2 \ln n^2} \le \left(\frac{1}{e}\right)^{c \ln n} \le \frac{1}{n^c}$$

Theorem: Algorithm KargerMinCut* runs in $O(cn^4 \log n)$ time and correctly reports a min-cut with probability at least $1-\frac{1}{n^c}$.

Improving KargerMinCut*

Theorem: Algorithm KargerMinCut* runs in $\overbrace{O(cn^4 \log n)}^{\text{not fast enough}}$ time and correctly reports a min-cut with probability at least $1 - \frac{1}{n^c}$.

$$\Pr\{C \text{ survives at time } n-1\}$$
$$\geq \left(1-\frac{2}{n}\right)\cdot\left(1-\frac{2}{n-1}\right)\cdot\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{3}\right) \geq \frac{1}{n^2}$$

$\frac{1}{3}$ too small!

$$\Pr\{C \text{ survives at time } n-n/\sqrt{2}\}$$
$$\geq \left(1-\frac{2}{n}\right)\cdot\left(1-\frac{2}{n-1}\right)\cdot\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{\lceil n/\sqrt{2}\rceil}\right)$$
$$= \left(\frac{n-2}{n}\right)\cdot\left(\frac{n-3}{n-1}\right)\left(\frac{n-4}{n-2}\right)\cdots\cdots\left(\frac{\lceil n/\sqrt{2}\rceil-2}{\lceil n/\sqrt{2}\rceil}\right) \geq \frac{1}{2}$$

Randomized Shrinking

Just make the problem smaller:

```
KargerSteinReduce(G)
    G_0 ← G
    for i=1 to |V| - ⌈|V|/√2⌉
        select a random edge (x_i, y_i) in E(G_{i-1})
        G_i ← G_{i-1} / (x_i, y_i)
    return G_i
```

Lemma: KargerSteinReduce runs in $O(n^2)$ time and produces a graph G' with $n/\sqrt{2}$ vertices. Furthermore, with probability at least $1/2$ $\text{mincut}(G') = \text{mincut}(G)$.

The Main Event

KargerSteinMinCut(G)
  if $|V| \leq 20$ then return MinCut(G)
  $G_1 \leftarrow$ KargerSteinReduce(G)
  $C_1 \leftarrow$ KargerSteinMinCut(G_1)
  $G_2 \leftarrow$ KargerSteinReduce(G)
  $C_2 \leftarrow$ KargerSteinMinCut(G_2)
  return $\min\{C_1, C_2\}$.

note: $G_1 \neq G_2$

Running Time: $T(n) = O(n^2) + 2 \cdot T(n/\sqrt{2}) = O(n^2 \log n)$

Prob. Correctness: $S(n) = \frac{1}{4} \cdot 0 + \frac{1}{2} \cdot S(n/\sqrt{2}) + \frac{1}{4} \cdot \left(1 - \left(1 - S(n/\sqrt{2})\right)^2\right)$

$$\geq \frac{1}{\log n}$$

Theorem: KargerSteinMinCut runs in $O(n^2 \log n)$ time and correctly outputs a min-cut of G with probability at least $1/\log n$

The Main Main Event

Theorem: KargerSteinMinCut runs in $O(n^2 \log n)$ time and correctly outputs a min-cut of G with probability at least $\overbrace{1/\log n}^{\text{too small}}$

KargerSteinMinCut*:
Run KargerSteinMinCut $c \cdot \log n \cdot \ln n$ times and output the best cut found

just right!

Theorem: KargerSteinMinCut* runs in $O(n^2 \log^3 n)$ time and correctly outputs a min-cut of G with probability at least $1 - \frac{1}{n^c}$.

Summary

Theorem: Let P be any maximization problem such that

1. Given an input S of size $O(1)$ we can compute $val(S)$ in $O(1)$ time
2. Given an input S of size n we can test if $val(S) \geq t$ in $D(n)$ time, for any real number t.
3. Given an input S of size n, we can partition S into subproblems $S_1, \ldots, S_r$ each of size at most αn and such that $val(S) = \min\{val(S_i) : i \in \{1, \ldots, r\}\}$

Then we can compute $val(S)$ in $O(D(n))$ expected time

Theorem: Given a graph G with n vertices, there exists an algorithm that runs in $O(cn^2 \log^3 n)$ time and outputs a cut of G that, with probability at least $1 - \frac{1}{n^c}$, is of minimum cardinality.