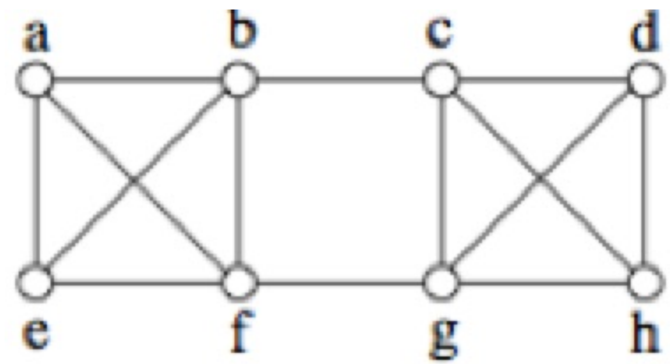
Karger’s Randomized min-cut algorithm.

Graphs (review of cse101)

- A graph consists of
  - a set of vertices V
  - a set of edges E



$$V = \{a,b,c,d,e,f,g,h\}$$

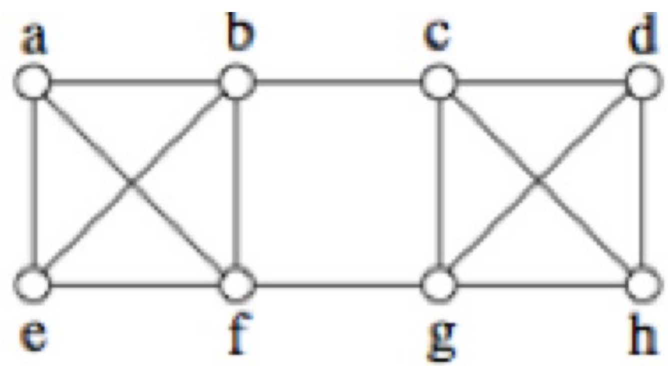
$$E = \begin{Bmatrix} (a,b),(a,f),(a,e), & (b,e),(b,f),(b,c), & (c,g),(c,h),(c,d), & (d,g),(d,h), \\ (e,f), & (f,g), & (g,h) & \end{Bmatrix}$$

$$n = |V|$$

$$m = |E|$$

Mutli Graphs

- A graph consists of
  - A set of vertices V
  - A bag of edges E
  - Still not cyclic edges (a,a)



$$V = \{a,b,c,d,e,f,g,h\}$$

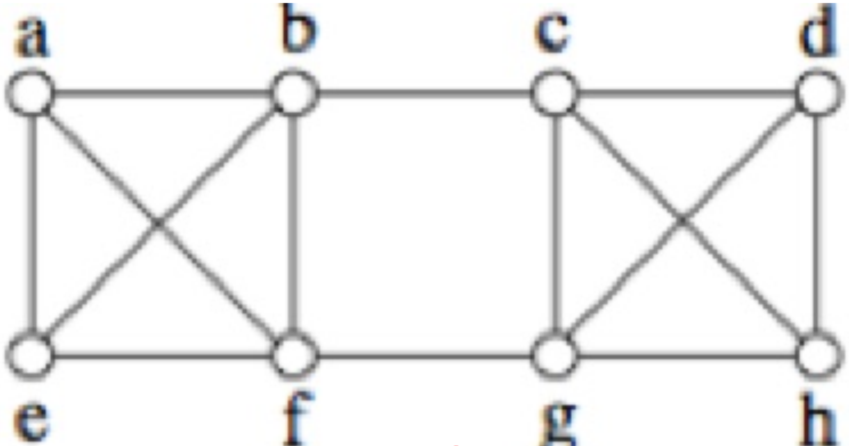
$$E = \left\{ \begin{matrix} (a,b),(a,f),(a,e), & (b,e),(b,f),(b,c), & (c,g),(c,h),(c,d), & (d,g),(d,h), \\ (e,f), & (f,g), & (g,h) & \end{matrix} \right\}$$

$$n = |V|$$

$$m = |E|$$

The min-cut problem

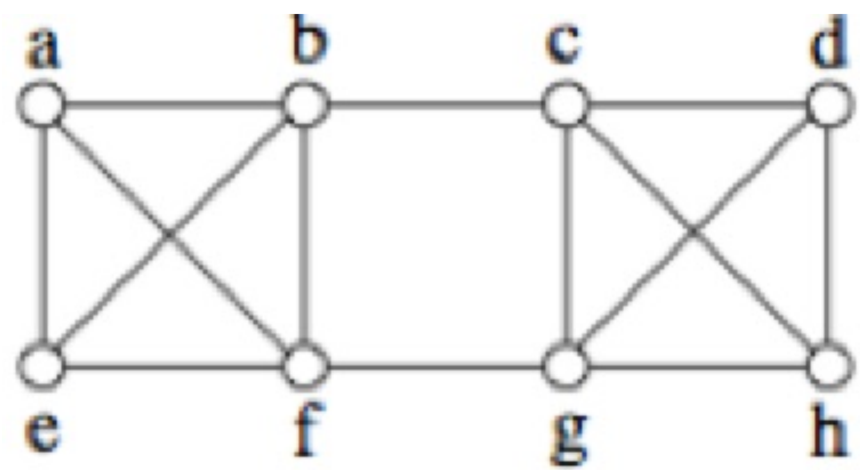
- Given a connected graph (V,E) find a partition of the vertices into two disjoint sets (V_1, V_2) such that k: the number of edges going between an element of V_1 and an element of V_2 is minimized.



K=2

Deterministic solution

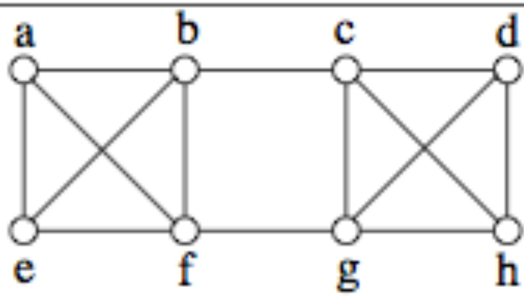
- Consider s-t min-cut between a fixed s and any other the other nodes.
- t-s min cut is equivalent to Max-Flow
- Solved using linear programming
- Time complexity : $O(nm \log(n^2/m))$



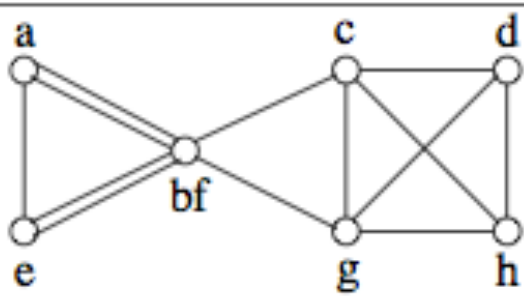
Karger's algorithm

- Choose an edge at random.
- Collapse the two nodes into one node.
  - Note that :
    - Collapsing two nodes decreases the number of nodes by 1.
- Repeat until only two nodes remains.
- The proposed min-cut is the partition of the original nodes defined by these two nodes.
- Count the number of edges across the cut.
- Repeat many times and take the minimal size cut.

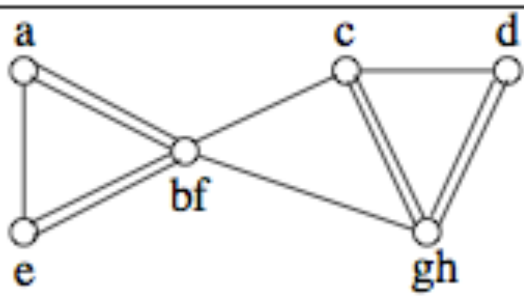
Example trace



14 edges to choose from
Pick $b - f$ (probability $1/14$)

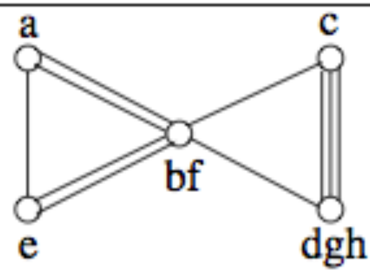
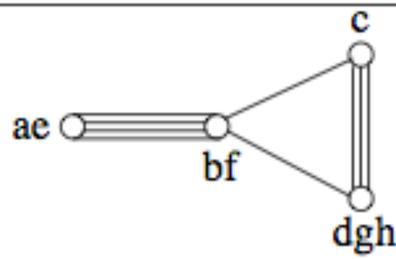
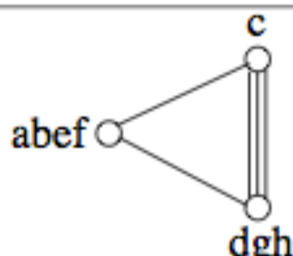


13 edges to choose from
Pick $g - h$ (probability $1/13$)



12 edges to choose from
Pick $d - gh$ (probability $1/6$)

Example trace (2)

	10 edges to choose from Pick $a - e$ (probability 1/10)
	9 edges to choose from Pick *ae-bf* (probability 4/9)
	5 edges to choose from Pick $c - dgh$ (probability 3/5)
	Done: just two nodes remain

When does karger's algorithm succeed?

- $\delta(S)$ the set of edges in a minimal cut.
- $k=|\delta(S)|$ is the size of the min cut.
- Karger's algorithm will find the min cut if none of the edges it selects belong to the min-cut set $\delta(S)$
- We will first upper bound the probability of hitting $\delta(S)$ in a single round.
- Then we will lower bound the probability of missing $\delta(S)$ in all rounds.

An upper bound on the probability of hitting the min-cut in one round

Each edge in the intermediate graph

corresponds to a single edge in the original graph

We want to upper bound the probability that the edge belongs to the min cut.

1) Order(node) = the number of edges connect to the node

2) Average Order$=\dfrac{2m}{n}$, because each edge contributes to the order of two nodes.

3) Min Order $\leq \dfrac{2m}{n}$

4) $k \doteq$ Min Cut $\leq$ Min Order $\leq \dfrac{2m}{n} \quad \Rightarrow \quad m \geq \dfrac{kn}{2}$

5) Prob. of choosing an edge in the min-cut $\leq \dfrac{k}{m} \leq \dfrac{2k}{kn} = \dfrac{2}{n}$

An lower bound on the probability of missing the min-cut set on all iterations

The number of vertices decreases by one after each iteration. Stopping when there are 2 nodes: n,n-1,n-2,.....,4,3 (the last edge is picked when there are 3 nodes remaining)

$$
\begin{aligned}
\Pr(\text{final cut is the minimum cut}) \quad &= \\
&= \Pr(\text{first selected edge is not in mincut}) \times \\
&\quad \Pr(\text{second selected edge is not in mincut}) \times \cdots \\
&\geq \left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\left(1-\frac{2}{n-2}\right)\cdots\left(1-\frac{2}{4}\right)\left(1-\frac{2}{3}\right) \\
&= \frac{n-2}{n}\cdot\frac{n-3}{n-1}\cdot\frac{n-4}{n-2}\cdots\frac{2}{4}\cdot\frac{1}{3} \\
&= \frac{2}{n(n-1)}.
\end{aligned}
$$

The running time complexity of Karger's algorithm

In order to boost the probability of success, we simply run the algorithm $\ell\binom{n}{2}$ times. The probability that at least one run succeeds is at least

$$1 - \left(1 - \frac{1}{\binom{n}{2}}\right)^{\ell\binom{n}{2}} \geq 1 - e^{-\ell}.$$

Setting $\ell = c \ln n$ we have error probability $\leq 1/n^c$. ■

It's easy to implement Karger's algorithm so that one run takes $O(n^2)$ time. Therefore, we have an $O(n^4 \log n)$ time randomized algorithm with error probability $1/\text{poly}(n)$.

A refined algorithm with an improved running time

Karger original algorithm was slightly worse than the state of the art. $O(n^4 \log(n))$ is worse than the running time of the min-cut/max-flow approach which is $O(nm \log(n^2/m)$ because m is at most n^2. In a later paper Karger and Stein proposed the following variant, which achieves expected running time of $O(n^2\log(n))$.

Improved algorithm: From a multigraph G, if G has at least 6 vertices, repeat twice:

1. run the original algorithm down to $n/\sqrt{2}+1$ vertices.
2. recurse on the resulting graph.

Return the minimum of the cuts found in the two recursive calls.

The choice of 6 as opposed to some other constant will only affect the running time by a constant factor.

We can easily compute the running time via the following recurrence (which is straightforward to solve, e.g., the standard Master theorem applies):

$$T(n) = 2\left(n^2 + T(n/\sqrt{2})\right) = O(n^2 \log n).$$

What is the highest value that the minimal degree can be?

- Suppose we have a graph with m edges and n nodes. Let k be the minimal degree of any node in the graph. Which of the following inequalities must hold?
- (1) $k \leq n-1$ (2) $k>0$ (3) $k \leq 2m/n$ (4) $k \leq m$

  A. 1,4
  B. 2,3
  C. 1,3,4
  D. 3

Prob of picking a min-cut edge

- The number of edges in the cut is at most the smallest degree in the graph $\leq 2m/n$
- If we pick one of the m edges at random the probability that we get an edge that is part of the cut is $(2m/n)/m = 2/n$
- Recall that the number of nodes decreases by 1 at each iteration.

How many times do we need to run Karger's algorithm

- The probability that Karger's algorithm is successful > $1/n^2$
- Say we run Karger's algorithm k times and use the smallest cut found. How large do we need to set k so that the probability that we find the min-cut is at least 1-ε ?

A. $(1-\epsilon)^k \leq \frac{1}{n^2}$ B. $\frac{k}{n^2} \geq 1-\epsilon$

C. $\left(1-\frac{1}{n^2}\right)^k \leq \epsilon$