Randomized Algorithms

Cut sparsification I: uniform edge sampling

Alex Gittens
Department of Computer Science, Rensselaer Polytechnic Institute

Fall 2026

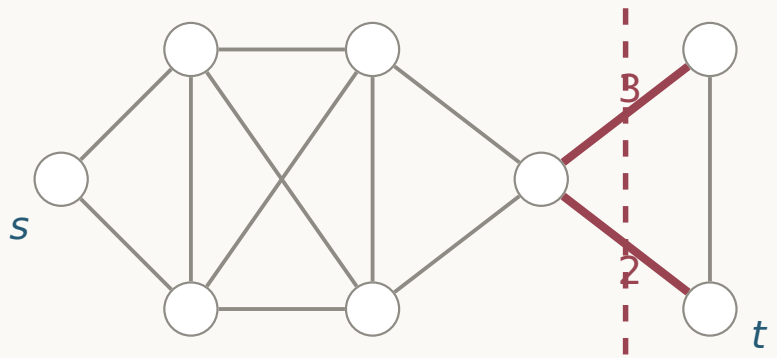
PART 1

The object we want

1

A smaller network for a maximum-flow computation

A network has n vertices, $m \gg n$ edges, and a capacity $w_e \geq 0$ on each edge. We want to approximate the maximum flow between fixed terminals s and t.



Unlabeled edges have capacity 1. The dashed line shows one cut, not necessarily a minimum cut.

Could we replace the network by a much smaller weighted graph and solve the flow problem there? The original graph stays fixed; we are compressing it, not changing its capacities.

Maximum flow is a minimum over cuts

For fixed terminals s, t, let $\mathcal{C}_{s,t} = \{S \subset V : s \in S,\ t \notin S\}$. By max-flow/min-cut,

$$F_G(s,t) = \min_{S \in \mathcal{C}_{s,t}} c_G(S).$$

Suppose a smaller weighted graph H approximates *every* cut of the fixed network:

$$(1-\varepsilon)c_G(S) \le c_H(S) \le (1+\varepsilon)c_G(S), \qquad S \subset V.$$

Take the minimum over $S \in \mathcal{C}_{s,t}$ in each term:

$$\begin{aligned}(1-\varepsilon)\min_{S \in \mathcal{C}_{s,t}} c_G(S) &\le \min_{S \in \mathcal{C}_{s,t}} c_H(S) \\ &\le (1+\varepsilon)\min_{S \in \mathcal{C}_{s,t}} c_G(S).\end{aligned}$$

Thus, for *every* choice of terminals in the same network,

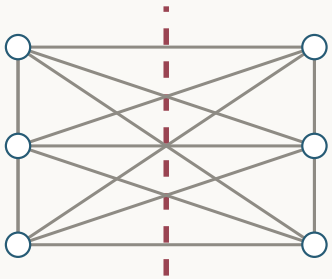
$$\boxed{(1-\varepsilon)F_G(s,t) \le F_H(s,t) \le (1+\varepsilon)F_G(s,t).}$$

The approximation is to the maximum-flow *value*; a flow on H is not automatically a feasible flow on G.

Why not keep a spanning tree?

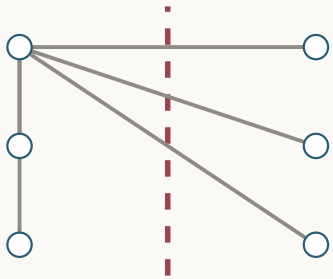
A spanning tree retains only $n - 1$ edges. But it can change the capacity of a cut substantially. Take the cut S separating three vertices of K_6 from the other three.

Clique K_6



$c_G(S) = 9$

Star spanning tree



$c_T(S) = 3$

The star keeps only three of the nine edges crossing this cut. We would like a sparse *weighted* graph that approximates every cut, not necessarily a tree.

Method	Number of edges	Construction
Spanning tree	$n - 1$	deterministic; poor cut approximation
Benczúr–Karger	$O(n \log n/\varepsilon^2)$	randomized sampling
BSS sparsifier	$O(n/\varepsilon^2)$	deterministic, stronger guarantee

What should a smaller graph preserve?

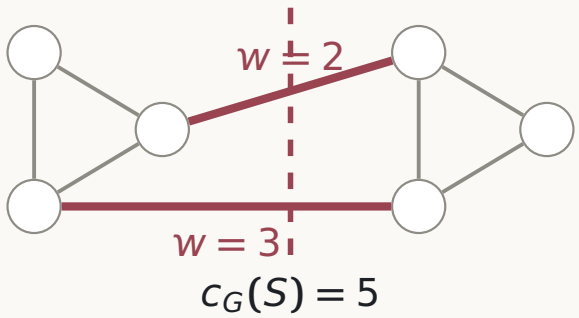
For a graph $G = (V, E, w)$ with nonnegative edge capacities, define

$$\delta_G(S) = \{uv \in E : |\{u, v\} \cap S| = 1\}, \qquad c_G(S) = \sum_{e \in \delta_G(S)} w_e.$$

We want a weighted graph H on the *same vertices*, with far fewer edges, satisfying

$$\boxed{|c_H(S) - c_G(S)| \le \varepsilon c_G(S) \quad \text{for every } S \subset V.}$$

There are $2^{n-1} - 1$ distinct nontrivial cuts. One graph H must approximate them all simultaneously.



Sample edges, then rescale their weights

We will sample edges independently and increase the capacity of each edge that survives. How should we choose its new capacity? If edge e has capacity w_e, survives with probability p_e, and receives capacity a_e when retained, requiring the expected capacity to equal w_e gives

$$p_e a_e = w_e \quad \Longrightarrow \quad a_e = \frac{w_e}{p_e}.$$

Thus unbiasedness holds by construction. An edge of capacity 2 sampled with probability $1/2$ is deleted or retained with capacity 4:

$$\mathbb{E}\widetilde{w}_e = \tfrac{1}{2}(0) + \tfrac{1}{2}(4) = 2.$$

By linearity, $\mathbb{E}c_H(S) = c_G(S)$ for each cut. But a realized sample need not be close to its expectation; we must control the random error for every cut at once.

PART 2

Warm-up: one probability for every edge

2

Uniform sampling: one probability for every edge

Suppose every edge has unit capacity. We independently retain each edge with probability p and give every retained edge weight $1/p$. For a cut S with $c = c_G(S)$ crossing edges, let Y_S count the edges retained. Then

$$Y_S = \sum_{i=1}^{c} X_i, \qquad X_i \overset{\text{iid}}{\sim} \text{Bernoulli}(p), \qquad Y_S \sim \text{Binomial}(c, p).$$

The weight of the sampled cut is $c_H(S) = Y_S/p$, with $\mathbb{E}c_H(S) = c$. For example, if $c = 4$ and $p = 1/2$, the sampled value is $2Y_S$ and may be 0, 2, 4, 6, or 8. We want to bound the probability that it differs from 4 by more than a chosen relative error.

How accurate is one sampled cut?

Start with the failure event we wish to bound:

$$\Pr\big(|c_H(S) - c| > \varepsilon c\big) = \Pr\left(\left|\frac{Y_S}{p} - c\right| > \varepsilon c\right)$$
$$= \Pr\big(|Y_S - pc| > \varepsilon pc\big).$$

Since $Y_S \sim \mathrm{Binomial}(c, p)$, multiplicative Chernoff gives, for $0 < \varepsilon < 1$,

$$\boxed{\Pr\big(|c_H(S) - c| > \varepsilon c\big) \le 2\exp(-\varepsilon^2 pc/3).}$$

A cut with more crossing edges has a smaller relative-error probability, for the same p.

Choosing p to preserve every cut

We want *every* nontrivial cut to have relative error at most ε. There are $2^{n-1}-1$ distinct cuts. Let $\lambda = \text{MINCUT}(G) > 0$. Every cut has $c_G(S) \geq \lambda$, so by the union bound,

$$\begin{aligned}\Pr(\exists \text{ inaccurate cut}) &\leq \sum_{\text{distinct cuts } S} 2e^{-\varepsilon^2 p c_G(S)/3} \\ &\leq (2^{n-1}-1)2e^{-\varepsilon^2 p\lambda/3} \\ &< 2^n e^{-\varepsilon^2 p\lambda/3}.\end{aligned}$$

To make this at most δ, it suffices that

$$\boxed{p\lambda \geq \frac{3(n\ln 2 + \ln(1/\delta))}{\varepsilon^2}.}$$

If the right-hand side exceeds λ, we must use $p = 1$, which means we just keep the original graph, unsparsified!

Why this bound gives no compression

For a simple unweighted graph, $\lambda \leq n - 1$. But the condition we just derived is

$$p\lambda \geq \frac{3(n \ln 2 + \ln(1/\delta))}{\varepsilon^2} > n - 1 \qquad (0 < \varepsilon < 1,\ \delta < 1).$$

Thus the bound never certifies any sampling with $p < 1$. Where was the estimate wasteful? In the union bound, we replaced the actual cut size $c_G(S)$ by λ for *every* cut, although large cuts have exponentially smaller failure probabilities.

To improve the analysis, we need to count cuts according to their size.

Why count cuts by their size?

Our union bound treated every cut as if it had the minimum value λ. That loses useful information: by Chernoff, a cut with $c_G(S) = 10\lambda$ has a much smaller failure probability than a minimum cut.

Consider the clique K_n. Its minimum cut has $n - 1$ edges. Exactly n cuts have this size: they separate one vertex from the other $n - 1$ vertices (for $n \geq 3$). Yet the graph has $2^{n-1} - 1$ distinct cuts in total.

Rather than multiply the worst failure probability by the number of *all* cuts, we should count how many cuts have size at most $\alpha\lambda$ and apply the union bound separately to groups of cuts of similar sizes.

Question: How many cuts can have size at most $\alpha\lambda$?

PART 3

Repairing the uniform-sampling proof

3

How many cuts are near-minimum?

Write $\lambda = \mathrm{MINCUT}(G)$. For an integer $\alpha \geq 1$, consider

$$N_\alpha = \#\{\text{distinct cuts } S : c_G(S) \leq \alpha\lambda\}.$$

If N_α grows only polynomially in n for fixed α, we can union-bound separately over cuts of size about $\alpha\lambda$. **Counting idea.** Build a random experiment that outputs one cut. If each cut in our family is output with probability at least q, there are at most $1/q$ such cuts, since the probabilities of distinct outputs sum to at most one. We'll use Karger's contraction process to construct such an experiment. Fix one cut C with $c_G(C) \leq \alpha\lambda$, and first estimate the probability that random contractions leave it intact.

Preserving a fixed cut during contraction

Fix C with $c_G(C) \le a\lambda$ and contract uniformly random edges until only $2a$ supervertices remain (assume $2a < n$). Suppose r supervertices remain and no edge of C has been contracted. Every supervertex has degree at least λ, hence there are at least $r\lambda/2$ edges. At most $a\lambda$ of those edges cross C.

$$\Pr(\text{next contraction crosses } C \mid C \text{ survived}) \le \frac{a\lambda}{r\lambda/2} = \frac{2a}{r}.$$

So the chance of avoiding C at this step is at least $(r - 2a)/r$.

Probability that the cut survives

Multiply the conditional probabilities from $r = n$ down to $r = 2a + 1$:

$$\Pr(C \text{ survives}) \geq \prod_{r=2a+1}^{n} \frac{r-2a}{r}$$
$$= \frac{(2a)!(n-2a)!}{n!} = \frac{1}{\binom{n}{2a}}.$$

This lower bound holds for *each* cut C of size at most $a\lambda$. We will use it to count how many such cuts there can be.

Counting near-minimum cuts

We now construct the random experiment promised earlier. It produces one cut of the original graph.

Experiment:

1. Contract uniformly random edges until $2a$ supervertices remain.
2. Choose uniformly among the $2^{2a-1}-1$ nontrivial cuts of the contracted graph, and output the corresponding cut of G.

Every supervertex represents a set of original vertices, all placed on the same side of the output cut.

Fix C with $c_G(C) \le a\lambda$. If it survives contraction, no supervertex contains vertices from both sides of C, so step 2 can select it. Thus

$$\Pr(\text{output} = C) \ge \underbrace{\frac{1}{\binom{n}{2a}}}_{\text{survives}} \underbrace{\frac{1}{2^{2a-1}-1}}_{\text{selected}} \ge \frac{1}{\binom{n}{2a}2^{2a-1}}.$$

Consequently $N_a \le \binom{n}{2a}2^{2a-1} \le n^{2a}$.

Karger's cut-counting theorem

Counting near-minimum cuts

For every integer $\alpha \geq 1$, the number of distinct cuts of an unweighted graph G with value at most $\alpha\lambda$ is at most $n^{2\alpha}$, where $\lambda = \text{MINCUT}(G)$.

For $2\alpha < n$, this follows from the contraction experiment. The last inequality uses

$$\binom{n}{2\alpha} \leq \frac{n^{2\alpha}}{(2\alpha)!} \quad \text{and} \quad (2\alpha)! \geq 2^{2\alpha-1}.$$

If $2\alpha \geq n$, the bound is immediate: there are only $2^{n-1} - 1 \leq n^{2\alpha}$ distinct cuts altogether.

Union bound over cuts of similar sizes

Let $\lambda = \text{MINCUT}(G)$. Group cuts into ranges

$$j\lambda \le c_G(S) < (j+1)\lambda, \qquad j = 1, 2, \ldots$$

There are at most $n^{2(j+1)}$ cuts in the jth range, and each fails with probability at most $2e^{-\varepsilon^2 pj\lambda/3}$. Therefore

$$\Pr(\exists \text{ inaccurate cut}) \le 2\sum_{j\ge 1} n^{2(j+1)} e^{-\varepsilon^2 pj\lambda/3}.$$

For example, with $p\lambda \ge 3(d+6)\ln n/\varepsilon^2$, the sum is at most

$$2n^2 \sum_{j\ge 1} n^{-(d+4)j} < n^{-d} \qquad (n \ge 2, d \ge 1).$$

The required $p\lambda$ has dropped from order n to order $\log n$.

Uniform sampling: what the better analysis proves

For $d \geq 1$, choose

$$p = \min\left\{1, \frac{3(d+6)\ln n}{\varepsilon^2 \lambda}\right\}, \qquad \lambda = \text{MINCUT}(G).$$

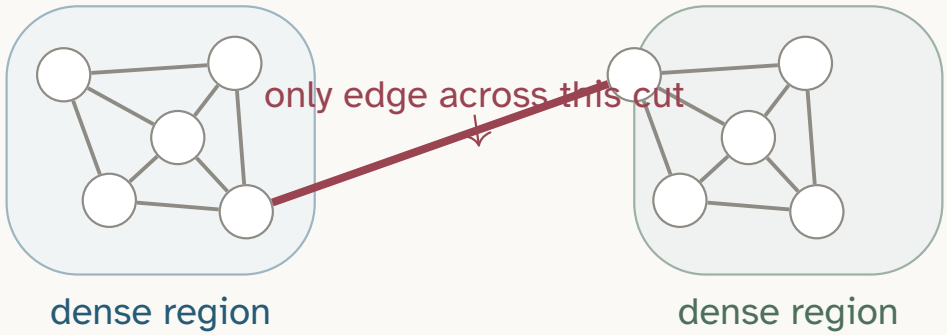
If $p < 1$, the preceding calculation shows that *all* cuts are preserved within relative error ε with probability at least $1 - n^{-d}$. If $p = 1$, preservation is exact. **How would we obtain λ?** We are assuming it is known while analyzing the sampler. A minimum-cut algorithm can compute it. The basic random contraction algorithm from the previous lecture can also find it with amplification, but is not a fast preprocessing method on large graphs.

A known lower bound on λ is safe in the denominator; an overestimate can cause undersampling.

The improved cut count makes the clique example possible. But the same global probability can still be wasteful for graphs with a small bottleneck.

A bridge forces uniform sampling to retain every edge

Connect two large cliques by a single bridge (only a few clique edges are drawn).



Here $\mathrm{MINCUT}(G) = 1$. Our uniform-sampling guarantee then uses $p = 1$: the bridge forces us to retain every edge, including the many edges inside the cliques. The issue is the *global* choice of p: the bridge needs special care, but that does not mean all clique edges should be sampled with the same probability.

What should nonuniform sampling achieve?

Suppose we choose a separate sampling probability p_e for each edge e, retaining it independently and giving it weight $1/p_e$.

1. **Keep small cuts accurate.** Edges that cross a cut of just a few edges cannot be discarded casually. A bridge, for example, must be kept.
2. **Retain few edges.** The expected number of edges in the output is

$$\mathbb{E}|E(H)| = \sum_{e \in E} p_e.$$

   We want this sum to be much smaller than m.
3. **Preserve all cuts at once.** Unbiasedness is not enough: one sampled graph must approximate every cut with high probability.

A useful restriction on an edge's sampling probability

A cut with c crossing edges was easy to control when pc was sufficiently large. With a separate probability p_e, a retained edge contributes $1/p_e$ to any cut it crosses. For an edge that is *not* retained deterministically, a natural restriction is to prevent its single random contribution from being too large relative to a cut it crosses:

$$\frac{1}{p_e} \leq \frac{c_G(S)}{\rho} \qquad \text{for every cut } S \text{ crossed by } e,$$

where ρ is on the order of $\varepsilon^{-2} \log n$. A cut with few edges imposes a stronger restriction than a large cut. This suggests choosing sampling probabilities according to how small a cut can contain an edge.

Next lecture: How can we measure that local connectivity, keep only $O(n \log n/\varepsilon^2)$ edges on average, and still prove that *all* cuts are accurate?