Isolating Cuts – Min Cut Differently

- Another way to use randomness to find min cuts in graphs
  – more recent, maybe generalizable in different ways.

- Basic Subroutine:

Min Isolating Cuts: given graph G, $\underset{\parallel}{(V,E)}$ "terminal" set $R \subseteq V$

Find cuts $(S_v, V \setminus S_v)$ for each $v \in R$
that is a minimum cut separating v from $R \setminus v$.

ie. $S_v = \arg\min_{S : S \cap R = \{v\}} |\partial S|$

Also works with non-negative edge weights, ignoring here.

Theorem: given (G, R) can find all R min isolating cuts in time needed to run $O(\log n)$ max-flows in some ~~n vertex~~ $|V|$ vertex graph and $|E|$ edge graph.

[Li Panigrahi '21]

[Since recent results show that max-flows can be done in near-linear time, this is also near-linear algo for isolating cuts].

Note: if $R = \{s,t\}$ then isolating cuts = min s-t cut.
if $R = V$ then isolating cuts are just "degree" cuts $S_v = \{v\}$.
But R being intermediate gives other structures.

Q1: How to prove above theorem?

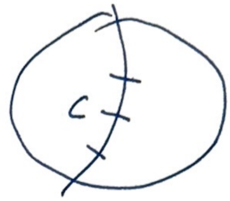
Q2: How does this help find global min cut?

Clearly can use $R = \{s,t\}$ ~~to~~ and enumerate over all t to find global min cut ~~with~~ ~~$O(n)$~~ $n-1$ isolating cuts.

Better?

Let's do Q2 first (uses randomization!)

Suppose ∂C is the minimum (global) cut in G.



Suppose also that $|C| \le |V \setminus C|$.

and it has k vertices.

Now suppose R is obtained by picking each vertex in V w.p. $1/k$ indep.

$$\Pr[|R \cap C| = 1] = \sum_{x \in C} \Pr[R \cap C = \{x\}]$$

$$\left(\tfrac{1}{k}\right)\left(1 - \tfrac{1}{k}\right)^{k-1} \approx \frac{(1/e)}{k}$$

Also works OK if we sample w.p. $\left[\frac{1}{2k}, \frac{2}{k}\right]$

$\Rightarrow$ sum $\sum$ is approx $1/e$. (constant prob)

$\Rightarrow$ if we repeat this experiment $\Theta(\log n)$ times,

$\Pr$(we fail each time) $\le 1 - 1/n^{10}$.

<u>Good</u>: so with high probability, $\exists$ one sample R s.t $|R \cap C| = 1$. (say $R \cap C = x$ for some x).

<u>Now</u>: C isolates x from $R \setminus x$ and is a global min-cut. So it is isolating cut too!

—— x ——

~~Then~~ But we don't know the size of C. (needed to sample @ rate $1/k = 1/|C|$)

$\boxed{\text{Try them all (in powers of 2).}}$

Constant factors change the probability of success only slightly.

So final Algorithm (Min Cut using Isolating Cuts)

for each size class $k = 1, 2, 2^2, 2^3, \cdots 2^{\log n}$

~~for each~~ for $\Theta(\log n)$ repetitions

~~find min~~ sample $R \leftarrow$ pick each vertex in V independently w.p. $1/k$.

find min isolating cuts for (G, R)

Return the smallest cut found.

Thm: Algo above finds min cut w.p. $1 - 1/\text{poly}(n)$.

Pf: for right choice of k, succeed w.p. $1 - 1/\text{poly}(n)$. in finding the min-cut.

$k \in [|C|, 2|C|]$ ∎

———— × ————

Great. How about finding min-isolating cuts? (Q1 above)

Very nice idea using the "submodularity" of the cut function.

Def: call a function $f: 2^V \to \mathbb{R}$ (which maps subsets of vertices to reals) submodular if

$$\boxed{f(A \cup B) + f(A \cap B) \leq f(A) + f(B)} \quad \forall A, B \subseteq V$$

Equivalent but more informative defn:-

$\forall A \subseteq B \subseteq V$ and $e \notin B$,

e adds more value to a subset A than to superset B.

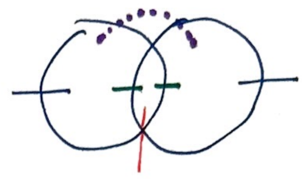
$$\boxed{f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)}$$

decreasing marginal returns.

Notation: Let $\boxed{d(S) = |\partial S|}$ # of edges crossing $(S, V \setminus S)$

if nonnegative weights $w_e \Rightarrow d(S) = \sum_{e \in \partial S} w_e$ also works.

Lemma 1: d is submodular.

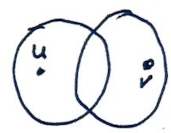


Pf: by picture. ☺

Lemma 2: d also satisfies $d(A) + d(B) \geq d(A \setminus B) + d(B \setminus A)$.

Pf: similar picture ☺

Lemma 3: can choose min isolating cuts $\{S_v\}_{v \in R}$ st the sets S_v are disjoint



Pf. Sps $S_u \cap S_v \neq \emptyset$.

also isolating cuts!

define $S_u' = S_u \setminus S_v$, $S_v' = S_v \setminus S_u$.

process is called "uncrossing"

(*) Submodularity says. $d(S_u) + d(S_v) \geq d(S_u') + d(S_v')$.

(**) because of minimality $d(S_u) \leq d(S_u')$, $d(S_v) \leq d(S_v')$

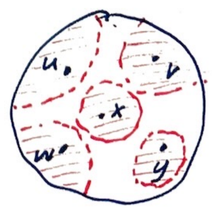
$\Rightarrow$ $d(S_u') + d(S_v') \geq d(S_u) + d(S_v)$

$\Rightarrow$ all must be equalities and hence S_u', S_v' also min isolating.

Repeat until all disjoint ▨

Great: this shows that we can describe

min isolating cuts using only near-linear # of bits



But: can we find them fast?

Yes. Clever use of submodularity!

Fix inclusion-wise smallest isolating cuts

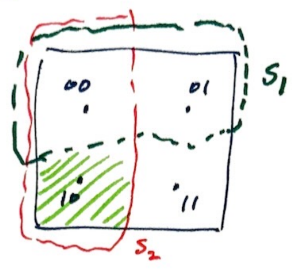
$\{S_x^*\}_{x \in R}$

Label
~~Write~~ vertices in R using $\log R$ bits

Let $R_i = \{v \in R \mid i^{\text{th}}$ bit of v's label has $1\}$

Find $C_i \subseteq V$ such that ∂C_i is minimum cut separating R_i from $R \setminus R_i$

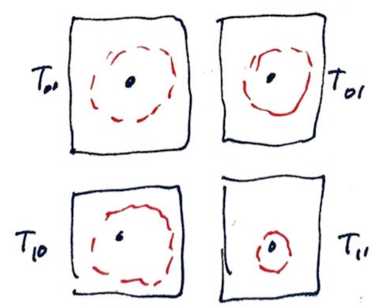
Delete all these edges.



Causes all these terminals to be separated from each other.

Let T_x be piece containing $x \in R$

(so T_{10} is marked in green highlight).



<u>Claim</u>: $S_x^* \subseteq T_x$. $\forall x \in R$

(S_x^* is smallest min isolating cut sep. x from $R \setminus x$)

<u>Pf</u>: sps not.

① S_x^* is min isolating cut so $d(S_x) \leq d(T_x)$.
T_x is some isolating cut

Assume that $x \in C_i$ $\forall$ iterations i (else flip C_i and $V \setminus C_i$).

If $S_x^* \subseteq C_i \ \forall i \Rightarrow S_x^* \subseteq T_x$.

So suppose $S_x^* \nsubseteq C_i$.

<u>Now</u>: $S_x^* \cap C_i$ also isolating. since S_x^* <u>minimal</u> $\Rightarrow d(S_x^* \cap C_i) \geq d(S_x^*)$

but $d(S_x^*) + d(C_i) \geq d(S_x^* \cap C_i) + d(S_x^* \cup C_i)$

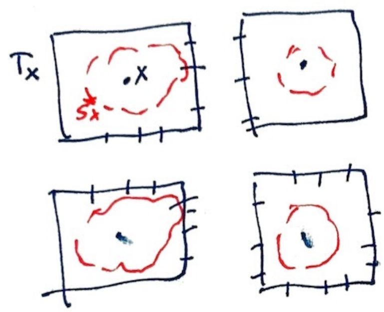
$\Rightarrow d(S_x^* \cup C_i) < d(C_i)$

but $S_x^* \cup C_i$ contains the same terminals as C_i (i.e. R_i)

which would violate the fact that C_i is a min cut

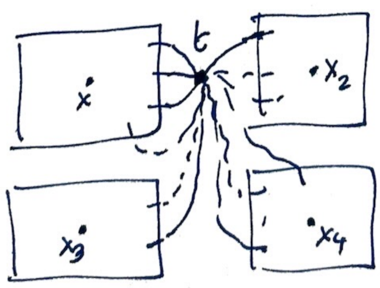
∎

<u>Great</u>: the picture looks like



So now just need to separate x from the boundary of T_x. for each $x \in R$.

Can just combine into one graph with $O(|E|)$ ~~extra~~ extra edges.



And run ~~shortest~~ max-flow/min-cut on this combined instance!

— x —

<u>Good</u>: to recap

deterministic!

(1) can find min isolating cuts in $O(\log n)$ max-flows!

randomized!

(2) can use isolating cuts routine $O(\log^2 n)$ times to find global min cut! (whp)