Cut problems:

Two of the long-standing open problems were
(1) Multicut — do a simpler version
(2) Directed Feedback Vertex Set — do the main ideas.

These (as Venkat said) were solved using a technique called "Important Cuts".
We'll talk about these today.

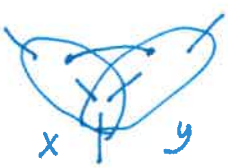
Prelims: (X,Y) disjoint vertex sets. Assume all unit edge capacities in $(V,E)=G$

· Max Flow/Min Cut Thm (Ford Fulkerson) If G has a set of k edges that separate X,Y then one can be found in $O(k(m+n))$ time. Else can find $k+1$ edge disjoint paths from X to Y.

Submodularity of Cut function:

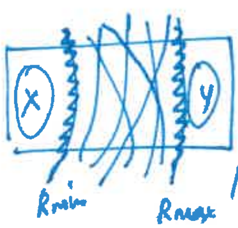
$\partial_G(X)$ = set of edges with exactly 1 endpt in X. $f(X) := |\partial_G(X)|$.

Submodular: $f(X)+f(Y) \geq f(X\cup Y)+f(X\cap Y)$. or decreasing marginal returns.



Cor: if $\partial X, \partial Y$ are st mincuts then so are $X\cup Y$ and $X\cap Y$.

So can define a maximal s-t cut and a minimal one. (Can find using FF).



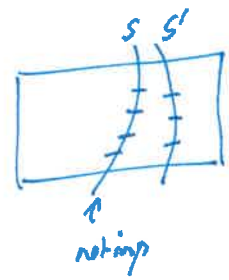
In general (X,Y) cuts, $\exists$ R_{max} R_{min} so that for any other cut $C\subseteq E$ sep X,Y, reachable vertices from X (say R_C) satisfy $R_{min} \subseteq R_C \subseteq R_{max}$.

Important Cuts: for any G, (X,Y), λ = min (X,Y)-cut size.

Sps $S\subseteq E$ cut, $R_S = R$ = reachable from X.

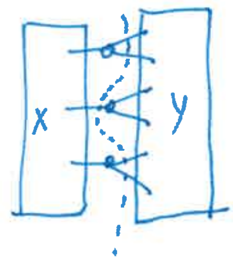
then S is important if (1) S is minimal (inclusion wise)

and (2) there is no S' st $|S'|\leq|S|$ and $R_{S'} \supsetneq R$. ← "rightmost of its size"

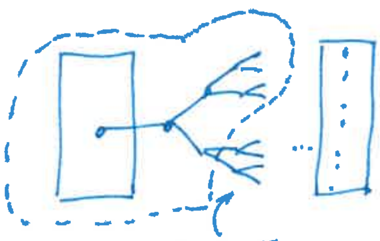


Fact: if $|S| = \lambda$ then unique important cut. (corr to R_{max}).

but could have many imp cuts otherwise



Fact: (from def^n).
$\forall S$ cuts $\exists S'$ important cut s.t.
$|S'| \leq |S|$
and $R_{S'} \supseteq R_S$.



imp cuts of size k = # of rooted subtrees of n-leaf binary tree with k leaves = Catalan # C_k.

$$= \frac{1}{k}\binom{2k-2}{k-1} \approx \Theta\left(\frac{2^{2k}}{k^{3/2}}\right) = \frac{4^k}{k^{3/2}}$$

Thm1: #(important cuts of size $\leq k$) is at most 4^k.

Thm2: Let $\mathcal{C}$ be set of important cuts in G (X,Y) always then $\sum_{C \in \mathcal{C}} 4^{-|C|} \leq 1$. [Implies Thm 1].

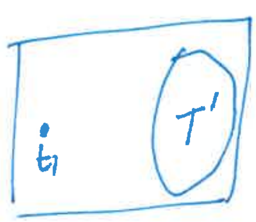
Pf: soon.

Multiway Cut: G, terminal set $T = \{t_1, t_2 .. t_k\}$. delete S s.t. there is no t_i, t_j path.
min # of edges. parameter = size of OPT $:= k$
[n.b. if param $= k$, problem is NP hard for $k=3$].

Thm [Marx]: $\exists$ $O(4^k \text{poly}(n))$ algo for Multiway Cut.

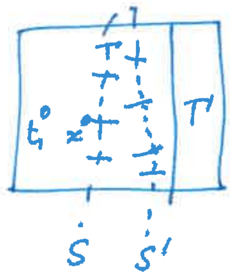
Warmup: $O^*(4^{k^2})$.

Consider t_1, $T \setminus t_1 =: T'$



~~[illegible]~~

[Pushing lemma] $\exists$ an opt solution s.t. it contains a (t_1, T')-imp. cut.



So push S to S' s.t. $R_{S'} \supseteq R_S$. $|S'| \le |S|$.
↑ important.

Claim: $(OPT - S) \cup S'$ also valid multiway cut

Pf: clearly seps t_1 from the others. Also, for t_i, t_j any uncut path must use S.
but then also cut by S'. ∎

So: Branch on which important cut. (at most 4^k of them).

⇒ recursion tree has branching at most 4^k. depth at most k

⇒ 4^{k^2} leaves. $O^*(4^{k^2})$ time.

Do a bit better: Claim: size of recursion tree $\le 4^k$ (leaves)

Note: If $\mathcal{S}_k$ = set of size-k important cuts then $\sum'_{C \in \mathcal{S}_k} 4^{-|C|} \le 1$ [by Thm 2]

~~But~~ By induction size of tree (#leaves) $\le \sum'_{C \in \mathcal{S}_k} 4^{k-|C|} = 4^k \cdot \sum'_{C \in \mathcal{S}_k} 4^{-|C|} \le 4^k$. ☺.

⇒ total size $\le k4^k$. ⇒ time $O^*(4^k)$. ∎

Multicut [was a Challenge problem]

Given G, $(s_1, t_1) \ldots (s_\ell, t_\ell)$ pairs. Now must separate s_i from its own t_i.

[Multiway cut ⇒ pairs are $\binom{T}{2}$ = all t_i from t_j].

Param say both k = size of sol^n, ℓ = # pairs.

Guess the connected components $\le (2\ell)^{2\ell}$ ways. Contract those terminals together.

Get multiway cut instance. Solve in $O^*(4^k)$ time.

⇒ $O^*(\ell^\ell 4^k)$.

Param = ℓ. impossible (NP hardness when $\ell = 3$). Param: $k + \ell$ [Marx]

Param = k? Finally shown by Marx & Razgon, and Bousquet Daligault Thomassé

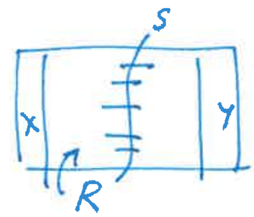
using random sampling & ~~...~~ important seps. [Won't be able to do today ☹]

OK. back to important separators:

let's show how to prove thm 2.

• Lemma: Given a (X,Y) cut S, can check if S is important cut.

Pf: Let R = reachable set.



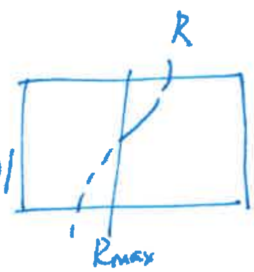
then S is imp cut iff $S = \partial R$ and ∂R is the unique minimum (R,Y)-cut.

Check in $O(|\partial R|(m+n))$ time ∎

• OK, remember: R_{max} is unique imp cut of size λ = mincut.
But may have many others.

Obs #: if S is an imp (X,Y) cut then
(a) $\forall e \in S$, $S-e$ is an imp (X,Y) cut in $G-\{e\}$.
(b) S is (X',Y) cut also then S is imp (X',Y) cut.
for $X' \supseteq X$.

• Lemma: ∂R is imp cut then $R_{max} \subseteq \partial R$.

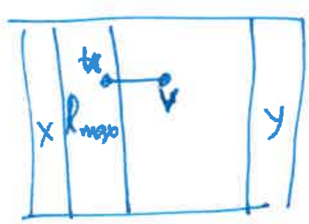


Pf: Sps not. $\underbrace{|\partial R_{max}|}_{=\lambda} + |\partial R| \geq \underbrace{|\partial(R_{max} \cap R)|}_{\geq \lambda} + |\partial(R_{max} \cup R)|$

$\Rightarrow |\partial(R_{max} \cup R)| \leq |\partial R|$.

And ↗ contains R. So ∂R not important since it has an equally good cut to its right.

OK: so start with



Either uv ∈ imp cut or not.

If $(u,v) \in \partial R \Rightarrow$ find imp cut in $(G-\{u,v\}, X, Y)$.
the one we're searching for
or $(G-\{u,v\}, R_{max}, Y)$.

else "contract" (u,v) and find imp cut in $(G, R_{max} \cup \{v\}, Y)$

In first case: $\lambda' \leftarrow \lambda - 1$ and $k \leftarrow k-1$. $\Rightarrow$ $2k-\lambda \rightarrow 2k'-\lambda' = (2k-\lambda)-1$.

In second case: $\lambda' \leftarrow \lambda+1$ (note!!) $\Rightarrow 2k'-\lambda' = (2k-\lambda)-1$. again.
$k \leftarrow k$

$\Rightarrow$ Recursion gives $2^{(2k-\lambda)}$ leaves. $=$ $4^k/2^\lambda$ leaves!!

Can be a bit more careful for Thm 2:

Inductive Claim: $\sum'_{C\in\mathcal{C}} 4^{-|C|} \leq 2^{-\lambda}$.

So then: cuts in Case 1 $(\mathcal{S}_1)$ $\sum'_{C\in\mathcal{S}_1} 4^{-|C|} = \frac{1}{4}\sum'_{C'\in\mathcal{S}_1} 4^{-|C'|} \leq \frac{1}{4}\cdot 2^{-(\lambda-1)}$ (by IH.)

removed (**) $= 2^{-\lambda}/2.$

Case 2: $(\mathcal{S}_2)$ $\sum'_{C\in\mathcal{S}_2} 4^{-|C|} \leq 2^{-(\lambda+1)} = 2^{-\lambda}/2.$ (by IH.)

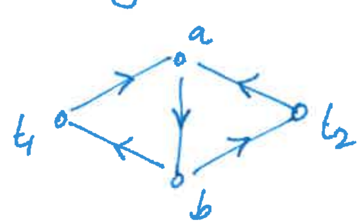
$+$

$2^{-\lambda}$. :)

Gives algorithm as well. [details in book if you want].

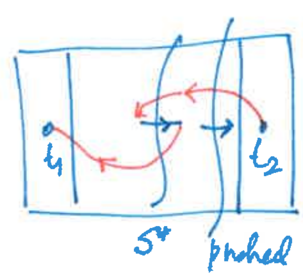
Extensions: A lot extends to directed cuts. (def^n of imp cuts, bound on # of cuts).

But: pushing lemma for multiway cut (directed) does not hold.



$k=1$ (edge ab) $\leftarrow$ unique.
but it is not important
and if push, get (b,t_2) or (b,t_1).

Why does this fail?



We may allow a $T\setminus\{t_1\} \rightarrow t_1$ path even though we still cut all $t_1 \rightarrow T\setminus t_1$ paths.

Still ~~E~~ we can solve Directed Multiway Cut (~~not disjointness~~) in FPT.
[see book for refs].

And we can solve this next problem

[Directed Skew Multicut]. Given $(s_1,t_1)\,(s_2\,t_2)\quad\ldots\quad(s_k,t_k)$

Delete edges so that s_1 cannot reach $t_1, t_2 \dots t_k$

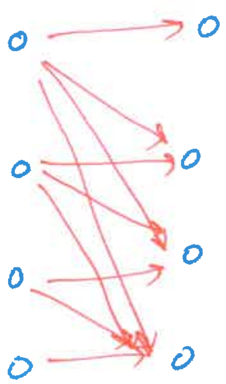
s_2 —— $t_2 \dots t_k$

⋮

s_j —— $t_j \dots t_k$

⋮

s_k —— t_k



red edge = separated.

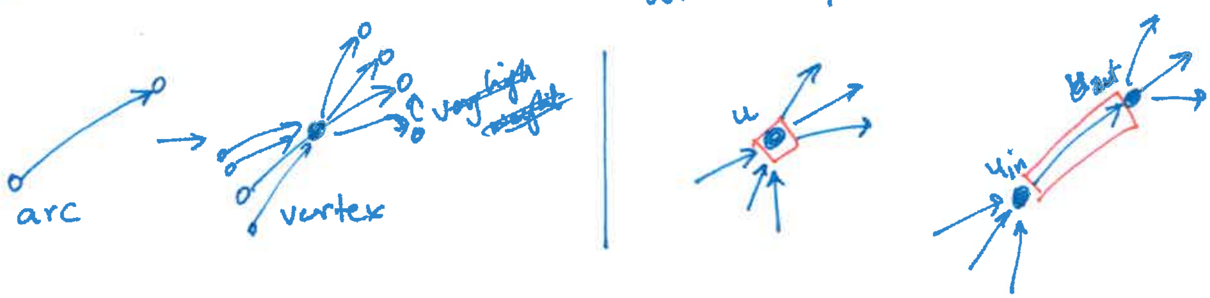
Same pf using probing Lemma.

gives FPT algo. in time $O^*(4^k)$.

OK: now can use Skew Multicut to solve Directed Feedback Vertex Set.

other challenge problem.

Fact 1: In directed graphs, (arc) edge version ≡ node ~~edge~~ version with same parameter.



Fact 2: Use iterative compression :-

[Given G and a $\boxed{\text{vertex}}$ solution of size k+1,

Want to find an $\boxed{\text{edge}}$ solution of size k.

feedback ~~vertex~~ arc set

(gives us disjointness trivially)

Say $W = \{w_1, w_2 \ldots w_{k+1}\}$.

So if $E^* \subseteq E$ is the optimal sol^n then $G \setminus E^*$ has no directed cycles.

So has a topological ordering, ~~such that~~ $v_1 \ldots v_n$

such no later vertex can reach an earlier one.

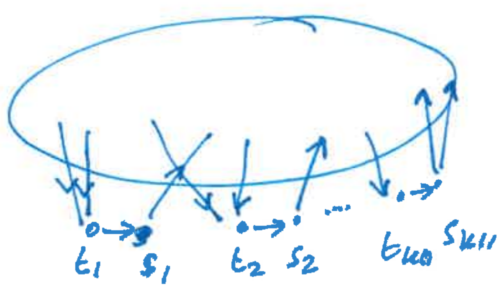
Restricted to W, this gives ordering on it. Say. $w_1\ w_2 \ldots w_{k+1}$.

Such that any directed cycle has to go ~~from~~ "backwards" at some pt.

i.e. $\exists$ portion $w_j \to w_i$ for $i \leq j$ (notice: $i=j$ is poss.) (then this portion is the empty transition)

Guess it! at most $((k+1)!)$ choices!

Split w_i into $t_i \to s_i$



Claim: $G \setminus E^*$ has acyclic ordering with $w_1 < w_2 < \ldots < w_{k+1}$

$\Downarrow$

E^* hits each $s_j \to t_i$ path for $j \geq i$

$\Updownarrow$

E^* is skew multicut on this graph.

$\Downarrow$

$G \setminus E^*$ is acyclic

Future Directions

- Optimal results

  show that $n^{\alpha} f(k)$ algos are possible and not possible to do better under ETH / SETH / GapETH / W[1] / P vs NP.

- Better results for nice graph classes.

  $2^{\sqrt{k}}$ algos for planar graphs.

- Approximation problems

  For classical problems, that elude better approximations, show can get better in FPT time?

  Eg: k-cut ~~2~~ 2apx in P, n^k ~~—~~ but W[1] hard.

  Otoh: get 5/3 in FPT(k). Get $(1+\varepsilon)$ in FPT(k, ε)?

  Eg: k-median 2.6 apx in P, W[1] hard.

  Get $1+2/e$ in FPT(k), – no better possible given GapETH.

  Eg: Steiner tree param by # of Steiner nodes.

  Get $(1+\varepsilon)$. Param by # terminals is in FPT.

- Algos in small space despite FPT runtime?

- And of course, better runtimes for classical problems.

Cut problems

Current best results (see, e.g. fpt.wikidot.com)

- Multiway cut. 4^k (we saw) $\to$ 2^k current best. $\boxed{\text{poly kernel not known}}$. know $O(k^{O(\ell)})$ OPT, # terminals
- Directed ~~Multiway~~ cut $2^{O(k^3)}$ no poly kernel
- Multicut $2^{O(k^3)}$ no $k^{O(1)}$ "
- Odd cycle deletion (aka MaxCut) $(2.3)^k$ poly kernel ~~for~~ w/ randomization (representative sets etc).
- Directed FVS $4^k.k!$ $\boxed{\text{poly kernel open}}$
- Directed Multicut W[1] hard parameterized by k, even when # terminal $\ell=4$.

- G, $S,T \subseteq V$ with min (S,T) cut of size r.
  $\Rightarrow \exists$ a set $Z \subseteq V$ of size $\le |S|.|T|.O(r)$. st.
  $\forall S' \subseteq S$, $T' \subseteq T$ Z contain a min (S',T') cut.
  Can find in randomized poly time (Monte Carlo).

  "cut sparsifier" graph compression

- Similar thm for multiway cuts.
