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 Vazirani said) were solved using a technique called “important cuts”.
We’ll talk about these today.

Relax: $(X,Y)$ disjoint vertex sets. Assume all unit edge capacities

$$\text{Max Flow/Min Cut Theory (Ford-Fulkerson).}$$

If $G$ has a set $S$ of edges that separates $X,Y$

then can be found in $O(k n)\text{ time. Else can find k edge disjoint paths from } X \text{ to } Y.$

Submodularity of Cut function:

$$\delta_G(S) = \text{ set of edges with exactly one endpoint in } S.$$  

$$f(S) = |\delta_G(S)|.$$  

Submodular:  

$$f(X) - f(Y) \geq f(X \cup Y) - f(X \cap Y).$$  

or marginal returns.

\begin{tikzpicture}
  \node (A) at (0,0) [circle, draw] {\(X\)};
  \node (B) at (2,0) [circle, draw] {\(Y\)};
  \draw (A) -- (B);
\end{tikzpicture}

Cor: if $\delta X, \delta Y$ are smallest then so are $X \cup Y$ and $X \cap Y$.

So can define a maximal set cut and a minimal one. (Confused why PF).

In general $(X,Y)$ cut, $3 R_\text{max} - R_{\text{min}}$ so that for any other cut $C \subseteq \text{Sep } X \text{,} Y$, reachable vertices from $X$ (say $R_\text{C}$) satisfy $R_{\text{min}} \leq R_\text{C} \leq R_{\text{max}}$.

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

Sp for $S \subseteq \text{ cut }$, $R_S = R = \text{ reachable from } X$.

then $S$ is important if

1. $S$ is minimal (indastic wise)
2. there is no $S' \text{ st } |S'| \leq |S|$

and $R_{S'} \neq R$. "rightmost of its size."
Fact: If $|S| = \lambda$ then unique important cut. (corr is $\lambda_{\text{max}}$).
but could have many imp cuts otherwise.

Fact: (Hoeffding).
Max cut $\exists$ opt cut s.t.
$|S'| \leq 1.5|S|$
and $R_{S} \geq 2R_{S}$.

$$\text{Imp cuts} = \# \text{ of rooted subtrees of n-leaf binary tree} = \text{Catalan} \# C_{k}$$
$$= \frac{1}{k^{2}} (\frac{2k}{k-1})^k \leq \Theta \left( \frac{2k}{k^2} \right) = \Theta \left( \frac{8k}{k^2} \right)$$

Thm: $\#(\text{important cuts of size } \leq k)$ is at most $4k$.

Thm2: Let $\mathcal{C}$ be set of important cuts in $G$ then
$$\sum_{C \in \mathcal{C}} \frac{1}{|C|} \leq 1.$$  [Implies Thm1].

**Proof:** soon.

**Multway Cut:** $G$, terminal set $T = \{t_{1}, t_{2}, \ldots, t_{k}\}$. delete $S$ s.t. there is no $t_{i}, t_{j}$ path.

min # of edges. parameter $= \text{size of OPT} = K$

[Claim: if parameter is $2$, problem is NP-hard for $k=3$.]

Thus [Mark]: $O(k^{2} \log(n))$ also for Multway Cut.

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

Consider $T, \overline{T} \cup t_{i} = T'$

**[Push this lemma]**: $\exists$ an opt solution s.t. it contains a $(t_{i}, T')$-imp cut.
So push $S_0.S'$ at $R_s/2R_s$. $|S'| \leq 15l$.

Claim: (OPT - $S$) $S'$ also valid multicut.

Proof: clearly no longer than the others. Also, for $t_0$, it's any path must use

but then also cut $S'$.

So: Branch on which important cut. (at most $4^k$ options.)

$\Rightarrow$ recursive here has branch at most $4^k$, depth at most $k$

$\Rightarrow 4^k$ leaves. $O(4^k)$ time.

Do it better: Claim: size of recursive here $\leq 4^k$ (leaves)

Note: If $S_k = S_0 + 4$ size-k important cuts then $\sum_{C \in S_k} 4^{-1C} \leq 1$ [by Thm2]

By induction, the size of tree $\leq \sum_{C \in S_k} 4^k-1C = 4^k \sum_{C \in S_k} 4^{-1C} \leq 4^k$. $\Rightarrow$

\[ \Rightarrow \text{total size } \leq k4^k. \Rightarrow \text{time } O^*(4^k). \]

has a [multicut challenge problem]

Given $G$, $(S_0, t_0)$ ... $(S_l, t_c)$ pairs. Now must separate & find $\mathbf{b_i}$.

[ multicut $\Rightarrow$ pairs are $(T)$ of all $t_i$'s in $T$ ]

Param: say both $k = \text{size of pool}$, $l = \text{# pairs}$.

Yields the connected components $\leq (2^e)^c$ ways. Use these branch and join.

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

$\Rightarrow O^*(4^k)$.

Param = $l$, impossible (NP hardness when $l = 3$). Param = $k + l$ [max]

Param = $k$? Foundly shown by Marx & Rat-ger, and Boncourt Dulant & Thomas.

Hard random sampling $\&$ important steps. [write be able to do today! 😊]
OK: back to important separators:

Let's show how to prove that.

- Lemma: Given a (x, y) cut S, one check if S is important cut.
  If: Let R = reachable set.

  Then S is imp cut iff S = DR
  and DR is the unique minimum

  Check in O(δ(G)δ(G)), triangle

- OK, remember: Rmax is unique imp cut & size λ = min cut.
  But may have many others.

  Obs 4: if S is an imp (x, y) cut then

  (a) y ∈ S, S - e is an imp (x, y) cut in G - e, e.

  (b) S is imp (x', y) cut also then S is imp (x', y) cut.
      for x' ∈ x.

- Lemma: DR is imp cut then Rmax ≤ DR.

  Pf: If not. |DRmax| + 1|DR| ≥ |DRmax(R)| + |DRmax(U)|

  = λ ≥ λ

  = |DRmax(U)| ≤ |DR|.

  And every contains R. So DR not important since it has an equally good cut as its pair.

OK: So start with

- Either UV is important or not
  - If UV ∈ DR ⇒ find imp cut in

    \[ (G - (U, V), x, y) \]

    the ones we're searching for or (G - (U, V), Rmax, y).

  - Else "contract" (x, y) and find imp cut in (G, Rmax, 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: \( \lambda' \leftarrow \lambda + 1 \)
\[ \Rightarrow 2k - \lambda' = (2k - \lambda) - 1. \text{ again.} \]
\[ k \leftarrow k \]
\[ \Rightarrow \text{Recursion gives} \ 2^{(2k - \lambda)} \text{ leaves} = 4^{k/2^\lambda} \text{ leaves}! \]

Can be a bit more careful for Thm 2:

**Inductive Claim:** \( \sum_{C \in \mathcal{E}} 4^{-|C|} \leq 2^{-\lambda}. \)

So then: auto in Case 1 (\( \mathcal{E}_1 \))
\[ \sum_{C \in \mathcal{E}_1} 4^{-|C|} = \frac{1}{4} \sum_{C \in \mathcal{E}_1} 4^{-|C|} \leq \frac{1}{4} \cdot 2^{-(\lambda - 1)} \]
removed (\( \mathcal{E}_1 \))
\[ = 2^{-\lambda} \]

Case 2: (\( \mathcal{E}_2 \))
\[ \sum_{C \in \mathcal{E}_2} 4^{-|C|} \leq \sum_{C \in \mathcal{E}_2} 2^{-(\lambda + 1)} = 2^{-\lambda} \]
by IH.
\[ = \]
\[ 2^{-\lambda} \]

**Extension:** A lot extends to directed cuts. (def: A \( \mathcal{I} \) imp cuts, band on \# \( \mathcal{I} \) cuts).

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

Why does this fail?

We may allow a \( T \setminus \{b\} \rightarrow \) 1 path

even though we still cut all \( t_i \rightarrow T \setminus t_i \) paths.
Still we can solve Directed Multicut \((n^2 \cdot \log(n))\) in \(FPT\).
[See book for refs.]

And we can solve this next problem

Directed

[Skew Multicut]. Given \((S_1, t_1, S_2, t_2, \ldots, S_k, t_k)\)

Delete edges so that \(S_i\) cannot reach \(t_1, t_2, \ldots, t_k\)

\(\overrightarrow{S_1} \quad \overrightarrow{t_1} \quad \overrightarrow{t_2} \quad \overrightarrow{t_k}\)

\(\overrightarrow{S_2} \quad \overrightarrow{t_2} \quad \overrightarrow{t_k}\)

\(\overrightarrow{S_k} \quad \overrightarrow{t_k}\)

Same pf using prob-lemma.

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

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

\(\overrightarrow{v_1} \quad \overrightarrow{v_2} \quad \overrightarrow{v_3} \quad \overrightarrow{v_4} \quad \overrightarrow{v_5}\)

Fact 1: In directed graphs, edge version \equiv node version with same parameter.

Fact 2: Use iterative compression.

Given \(G\) and a vertex solution of size \(k+1\),

Want to find an edge solution of size \(k\).

\[\text{feedback vertex set, arc}\]
Say \( W = \{ w_1, w_2, \ldots, w_{k+1} \} \).

So if \( E^* \subseteq E \) is the optimal solution then \( G \setminus E^* \) has no directed cycles.

So has a topological ordering, \( v_1, \ldots, v_n \), 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 "backwards" at some point.

i.e. I partition \( W_i \to W_j \) for \( i \leq j \) (notice: \( i = j \) is poss.)

Then this partition 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 \( W_1 < W_2 < \ldots < W_{k+1} \)

\( E^* \) hits each \( s_i \to t_i \) path for \( j \geq i \)

\( E^* \) is skew multicut on this graph.

\( G \setminus E^* \) is acyclic
Future Directions

- Optimal results: $n^k f(k)$ also are possible and not possible to do better under ETH/SETH/GapETH/WETH/PwNP.

- Better results for nice graph classes: $2\sqrt{k}$ algo for planar graphs.

- Approximation problems

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

- Eg: $k$-cut $\leq 2$axp in P, $n^k$ is but WETH hard.

- Otah: get $\frac{5}{6}$ in PPT($k$). Get $1+\varepsilon$ in $PPT(4\varepsilon)$?

  - Eg: $k$-median $2.6$ axp in P, WC(1) hard.

  - Get $1+\varepsilon$ in PPT($k$), -no better possible given ETH.

- Eg: Steiner tree pamam by $\#4$Steiner nodes.

- Get $1+\varepsilon$.

  - pamam by $\#4$terminals in in PPT.

- Kyos in small space despite PPT runtime?

- And of course, better runtimes for classical problems.
Cut problems

Current best results (see, e.g., ftwiki.dot.com)

- Multicut

\[ 4^k \text{ (best seen)} \rightarrow 2^k \text{ current best} \]

Poly kernel not known, known \( \Omega(2^{o(\sqrt{k})}) \)

- Directed Multicut

\[ 2^{O(k^2)} \] no poly kernel

- Multicut

\[ 2^{O(k^3)} \] no \( k^{O(1)} \)

- Odd cycle deletion (also MaxCut)

\[ (2.5)^k \] poly kernel + wij randomization (representative set etc.)

- Directed PVS

\[ 4^k \cdot k! \] poly kernel open

- Directed Multicut

Wij hard parameterized by \( k \), even when \( \# \text{terminal} \leq 4 \).

- \( G, S, T \subseteq V \) into min \((S, T)\) cut of size \( r \).

\[ \exists \text{ set } Z \subseteq V \text{ of size } \leq 1.111.000 \cdot St. \]

\[ \forall S \subseteq S, T' \subseteq T \text{ contain a min } (S', T') \text{ cut.} \]

Can find in randomized polynomial time (Monte Carlo).

- Similar then for multicut.