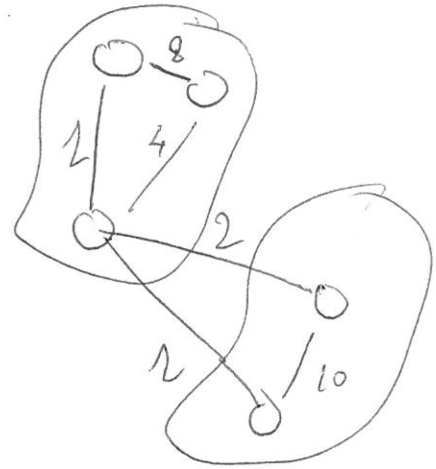
Min-Cut



Input: Graph $G=(V,E)$
weights $w: E \to \mathbb{R}^+$

Output Cut $C \subseteq V$ of min weight

$$\sum_{\substack{u \in C \\ v \notin C \\ (u,v) \in E}} w(u,v)$$

1. Find a polynomial time algorithm for min cut.

For every $s \neq t \in V$ find min s-t cut (via flow algorithm)
Pick the smallest cut of them.

$$\text{Run-time} \leq |V|^2 \cdot \underset{\text{run-time}}{\text{flow}}$$

$$\underset{\text{run-time}}{\text{flow}} \leq O(|V|+|E|) \cdot \underbrace{\sum_{e \in E} w(e)}_{\text{upper bound on max flow in } G}$$

Max-Cut

Input: Graph $G=(V,E)$

weights $w: E \to \mathbb{R}$

Output: Cut $C \subseteq V$ of max weight

(2) Show that max cut is NP-hard.

We'll show:

$$3SAT \le_p NAE\text{-}4SAT \le_p NAE\text{-}3SAT \le_p \text{Max-Cut}$$

Where

Not All Equal

NAE-kSAT

Input: ~~3SAT~~ formula $C_1 \wedge \dots \wedge C_m$ where over variables $x_1 \dots x_n$

$C_i \equiv NAE(\lambda_1, \lambda_2, \lambda_3)$ where each λ_i is a literal of the form $(\neg)x_j$

Output: Is there an assignment to $x_1 \dots x_n$ that satisfies $C_1 \wedge \dots \wedge C_m$.

I) $3SAT \leq_p NAE\text{-}4SAT$

we'll replace with a new global variable z

consider either $x_1 \ldots x_n, z$ (if z=1
or
$\neg x_1 \ldots \neg x_n, \neg z$ (if z=

Note that $\lambda_1 \vee \lambda_2 \vee \lambda_3 \equiv NAE(\lambda_1, \lambda_2, \lambda_3, \cancel{F})$.

Reduction replaces each 3SAT clause with the corresponding NAE clause. Runs in linear time.

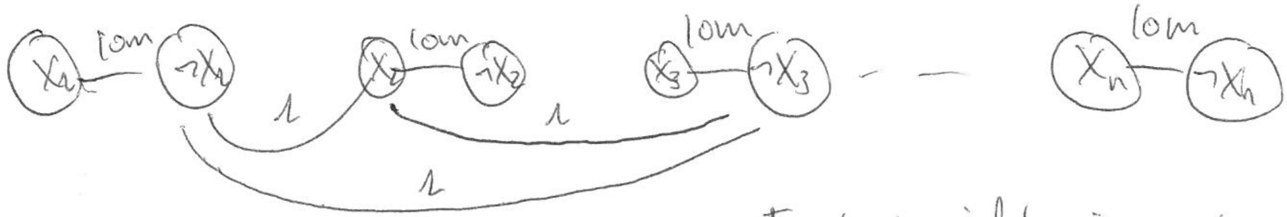
II) $NAE\text{-}4SAT \leq_p NAE\text{-}3SAT$

$$NAE(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \equiv NAE(\lambda_1, \lambda_2, w) \wedge NAE(\lambda_3, \lambda_4, \neg w)$$

where w is a new variable.

Reduction replaces each NAE-4SAT clause with the two corresponding clauses. Runs in linear time.

III) $NAE\text{-}3SAT \leq_p Max\text{-}Cut$



$NAE(\neg x_1, x_2, \neg x_3)$

Each variable is replaced with two vertices connected by an edge.

Each clause is replaced with a triangle.

Linear time reduction.

Claim NAE-3SAT formula is satisfiable $\Longleftrightarrow$ There is a cut of weight $\geq 10mn+2m$.

Pf $(\rightarrow)$ If formula is satisfiable, consider

$C =$ all literals that evaluate to true in satisfying assignment.

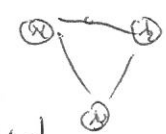
$$\text{Weight} = \underbrace{10mn}_{\substack{\text{edges}\\(x_i, \neg x_i)}} + \underbrace{2m}_{\substack{\text{two edges}\\\text{of each triangle}\\\text{are cut.}}}$$

$(\leftarrow)$ Consider a cut of weight $\geq 10mn+2m$.

* $\forall i$ x_i and $\neg x_i$ have to be on different sides of the cut - otherwise weight $\leq 10m(n-1)+3m$
$< 10mn + 2m$

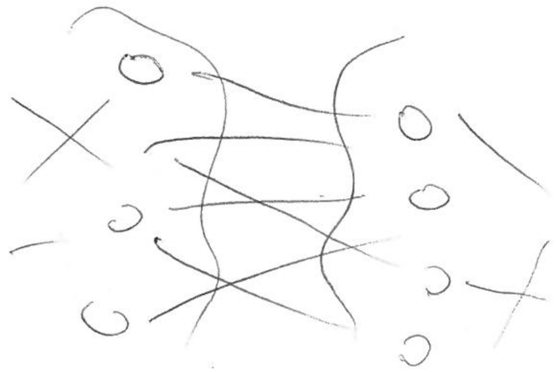
* Exactly two edges of every triangle cross the cut

Take an assignment that assigns TRUE to all the literals on one side of the cut.



This is well-defined and satisfies all NAE clauses.

3 How can Min-Cut be easy & Max-Cut be NP-hard?



$\forall e \in E \quad w(e) \mapsto -w(e)$

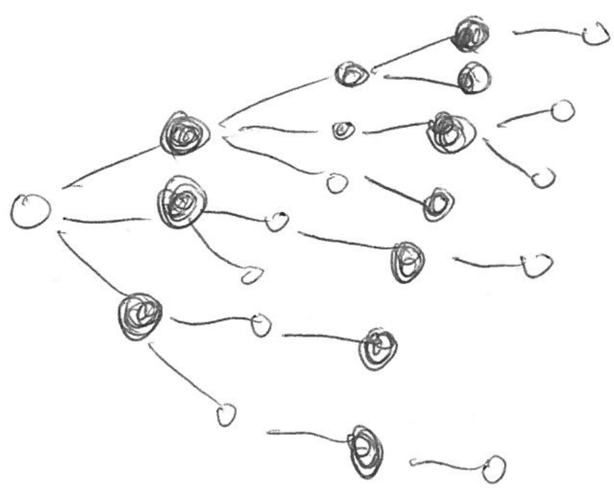
max-cut $\leftrightarrow$ min-cut

The flow algorithm for min cut requires non-negative weights!

The problem is NP-hard otherwise.

(4) Show a polynomial time algorithm for finding max cut on a tree.

Does this contradict the NP-hardness of Max-Cut?



the odd-even cut cuts

Observe: <u>all</u> the edges of a tree.

(similar to test 1 problem on 2-coloring)

NP-hardness doesn't rule out the existence of easy inputs - it only implies the <u>existence</u> of hard instances (assuming $P \neq NP$).