Computational Complexity Theory

Lecture 5: More NP-complete problems

Department of Computer Science,
Indian Institute of Science

Recap: 3SAT is NP-complete

- Definition. A CNF is a called a k-CNF if every clause has at most k literals.

  e.g. a 2-CNF $\phi = (x_1 \vee x_2) \wedge (x_3 \vee \neg x_2)$

- Definition. k-SAT is the language consisting of all *satisfiable k-CNFs.*

- Theorem. *(Cook-Levin)* 3-SAT is NP-complete.

NP complete problems: Examples

- Independent Set
- Clique
- Vertex cover
- 0/1 integer programming
- Max-Cut (NP-hard)

Karp 1972

- 3-coloring planar graphs *Stockmeyer 1973*
- 2-Diophantine solvability *Adleman & Manders 1975*

Ref: Garey & Johnson, "*Computers and Intractability*" 1979

NPC problems from number theory

- SqRootMod: Given natural numbers a, b and c, check if there exists a natural number $x \leq c$ such that

$$x^2 = a \pmod{b}.$$

- Theorem: SqRootMod is NP-complete.

Manders & Adleman 1976

NPC problems from number theory

- Variant_IntFact : Given natural numbers L, U and N, check if there exists a **natural number** $d \in [L, U]$ such that d divides N.

- Claim: Variant_IntFact is NP-hard under *randomized poly-time reduction*.

- Reference: *https://cstheory.stackexchange.com/questions/4769/an-np-complete-variant-of-factoring/4785*

A peculiar NP problem

- Minimum Circuit Size Problem (MCSP): Given the **truth table** of a Boolean function f and an integer s, check if there is a circuit of size $\leq$ s that computes f.

- Easy to see that MCSP is in NP.

- Is MCSP NP-complete? Not known!

A peculiar NP problem

- Minimum Circuit Size Problem (MCSP): Given the **truth table** of a Boolean function f and an integer s, check if there is a circuit of size $\leq$ s that computes f.

- Easy to see that MCSP is in NP.

- Is MCSP NP-complete? Not known!
- Multi-output MCSP is NP-hard under poly-time randomized reductions. *(Ilango, Loff, Oliveira 2020)*

A peculiar NP problem

- Minimum Circuit Size Problem (MCSP): Given the **truth table** of a Boolean function f and an integer s, check if there is a circuit of size ≤ s that computes f.

- Easy to see that MCSP is in NP.

- Is MCSP NP-complete? Not known!
- Partial fn. MCSP is NP-hard under poly-time randomized reductions. *(Hirahara 2022)*

More NP-complete problems

Example 1: Independent Set

- INDSET := {(G, k): G has independent set of size k}

- Goal: Design a poly-time reduction f s.t.

$$x \in 3SAT \Leftrightarrow f(x) \in INDSET$$

- Reduction from 3SAT: Recall, a reduction is just an efficient algorithm that takes input a 3CNF ϕ and outputs a (G, k) tuple s.t

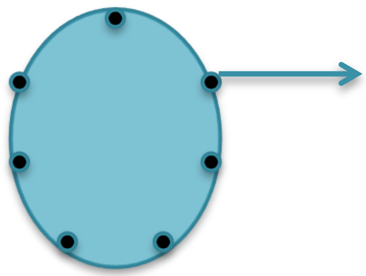
$$\phi \in 3SAT \Leftrightarrow (G, k) \in INDSET$$

Example 1: Independent Set

- Reduction: Let ϕ be a 3CNF with m clauses and n variables. Assume, every clause has exactly 3 literals.

Example 1: Independent Set

- Reduction: Let ϕ be a 3CNF with m clauses and n variables. Assume, every clause has exactly 3 literals.

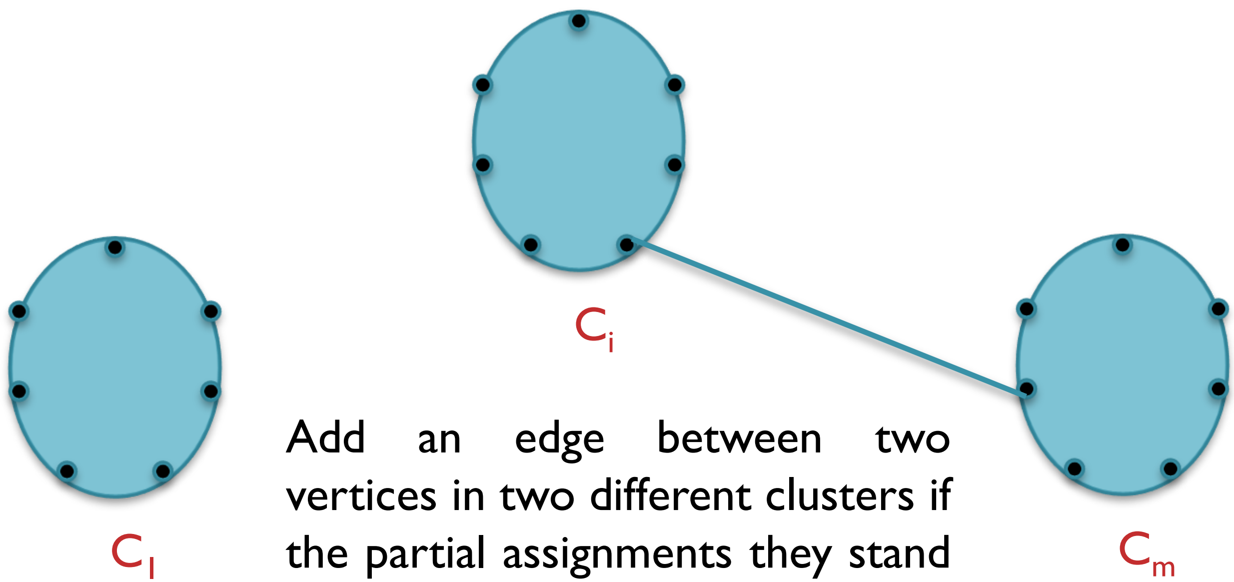


A vertex stands for a partial assignment of the variables in C_i that satisfies the clause

For every clause C_i form a complete graph (cluster) on 7 vertices

Example 1: Independent Set

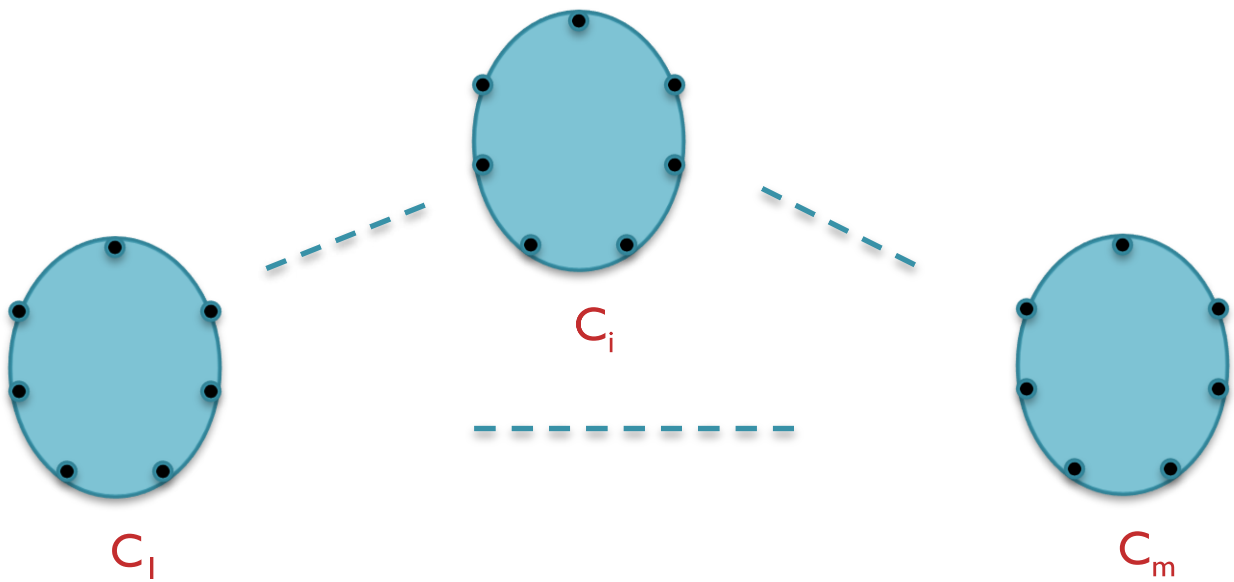
- Reduction: Let ϕ be a 3CNF with m clauses and n variables. Assume, every clause has exactly 3 literals.



Add an edge between two vertices in two different clusters if the partial assignments they stand for are <u>incompatible</u>.

Example 1: Independent Set

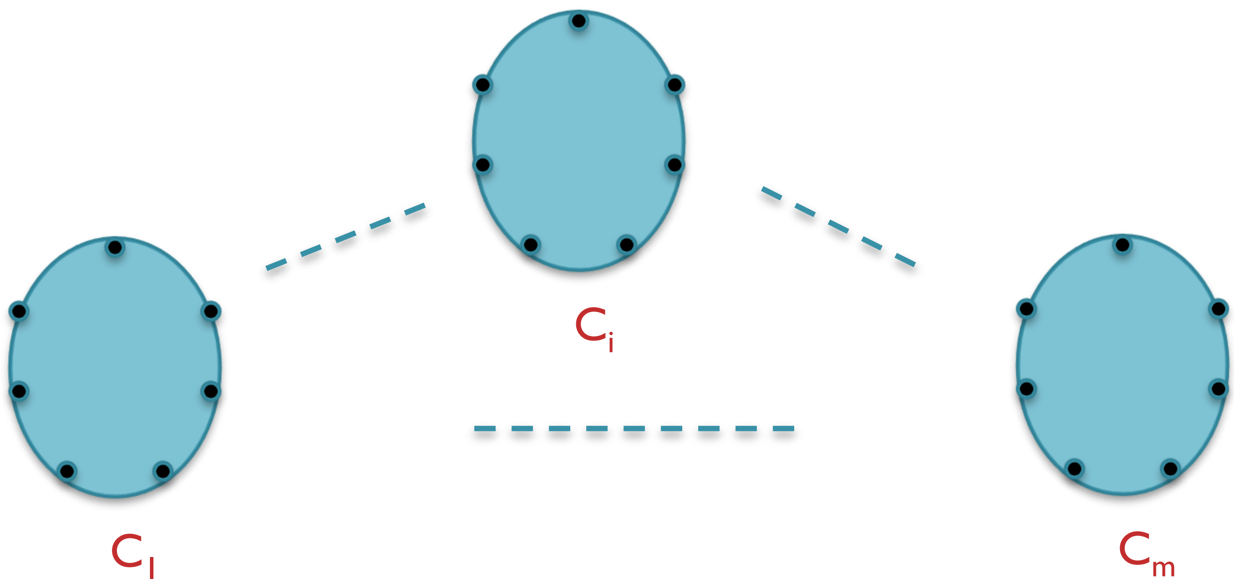
- Reduction: Let ϕ be a 3CNF with m clauses and n variables. Assume, every clause has exactly 3 literals.



Graph G on $7m$ vertices

Example 1: Independent Set

- Reduction: Let ϕ be a 3CNF with m clauses and n variables. Assume, every clause has exactly 3 literals.



- Obs: ϕ is satisfiable iff G has an ind. set of size m.

Example 2: Clique

- CLIQUE := {(H, k): H has a clique of size k}

- Goal: Design a poly-time reduction f s.t.

$$x \in \text{INDSET} \quad \Leftrightarrow \quad f(x) \in \text{CLIQUE}$$

- Reduction from INDSET: The reduction algorithm computes $\overline{G}$ from G

$$(G, k) \in \text{INDSET} \quad \Leftrightarrow \quad (\overline{G}, k) \in \text{CLIQUE}$$

Example 3: Vertex Cover

- VCover := {(H, k): H has a vertex cover of size k}

- Goal: Design a poly-time reduction f s.t.

  $x \in$ INDSET $\Leftrightarrow$ $f(x) \in$ VCover

- Reduction from INDSET: Let n be the number of vertices in G. The reduction algorithm maps (G, k) to (G, n-k).

  $(G, k) \in$ INDSET $\Leftrightarrow$ $(G, n-k) \in$ VCover

Example 4: 0/1 Integer Programming

- 0/1 IProg := Set of satisfiable 0/1 integer programs
- A <u>0/1 integer program</u> is a set of linear inequalities with rational coefficients and the variables are allowed to take only 0/1 values.

- Reduction from 3SAT: A clause is mapped to a linear inequality as follows

$$x_1 \vee \bar{x}_2 \vee x_3 \longrightarrow x_1 + (1 - x_2) + x_3 \geq 1$$

Example 5: Max Cut

- MaxCut : Given a graph find a <u>cut</u> with the max size.
- A <u>*cut*</u> of $G = (V, E)$ is a tuple $(U, V \setminus U)$, $U \subseteq V$. <u>Size</u> of a cut $(U, V \setminus U)$ is the number of edges from U to $V \setminus U$.

- MinVCover: Given a graph H, find a vertex cover in H that has the min size.

- Obs: From MinVCover(H), we can readily check if $(H, k) \in$ VCover, for any k.

Example 5: Max Cut

- MaxCut : Given a graph find a <u>cut</u> with the max size.
- A *cut* of $G = (V, E)$ is a tuple $(U, V \setminus U)$, $U \subseteq V$. <u>Size</u> of a cut $(U, V \setminus U)$ is the number of edges from U to $V \setminus U$.

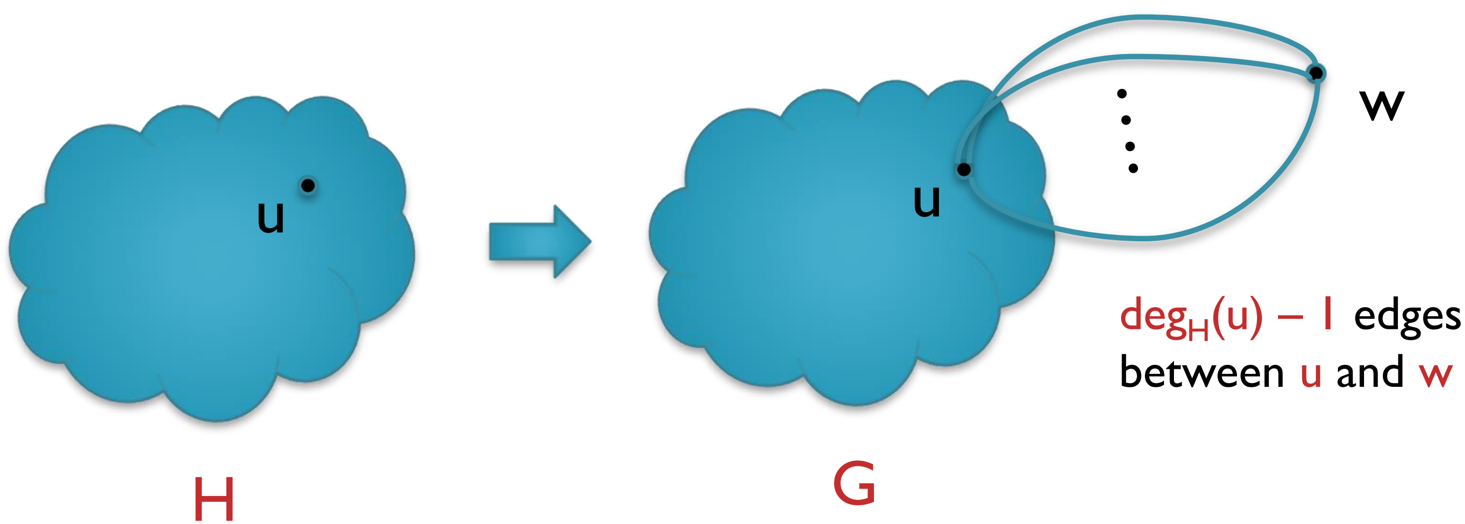
- Goal: A poly-time <u>reduction</u> from MinVCover to MaxCut.

$$H \xrightarrow{f} G \quad \text{s.t.}$$

$$\boxed{\text{Size of a MaxCut}(G) = 2.|E(H)| - |\text{MinVCover}(H)|}$$

Example 5: Max Cut

- The reduction: H $\xrightarrow{f}$ G



- G is formed by adding a new vertex w and adding $\deg_H(u) - 1$ edges between every $u \in V(H)$ and w.

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$

Example 5: Max Cut

- Claim: $|\mathrm{MaxCut}(G)| = 2.|E(H)| - |\mathrm{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \setminus U + w)$ is a cut in G.

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Let $S_G(U)$:= no. of edges in G with exactly one end vertex incident on a vertex in U.

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Let $S_G(U)$ = no. of edges going out of U in G.

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Let $S_G(U)$ = size of the cut $(U, V \backslash U + w)$.

Example 5: Max Cut

- Claim: $|MaxCut(G)| = 2.|E(H)| - |MinVCover(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Let $S_H(U)$:= no. of edges in H with <u>exactly one</u> end vertex incident on a vertex in U.

Example 5: Max Cut

- Claim: $|\mathrm{MaxCut}(G)| = 2.|E(H)| - |\mathrm{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V\backslash U + w)$ is a cut in G.

- Then $S_G(U) = S_H(U) + \sum_{u \in U} (\deg_H(u) - 1)$

$$= S_H(U) + \sum_{u \in U} \deg_H(u) - |U|$$

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \setminus U + w)$ is a cut in G.

- Then $S_G(U) = S_H(U) + \sum_{u \in U} (\deg_H(u) - 1)$

$$= S_H(U) + \sum_{u \in U} \deg_H(u) - |U|$$

Obs: Twice the number of edges in H with at least one end vertex in U.

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Then $S_G(U) = S_H(U) + \sum_{u \in U} (\deg_H(u) - 1)$

$$= S_H(U) + \sum_{u \in U} \deg_H(u) - |U|$$

$$= 2.|E_H(U)| - |U|$$

$E_H(U)$:= Set of edges in H with <u>at least one</u> end vertex in U.

Example 5: Max Cut

- Claim: $|MaxCut(G)| = 2.|E(H)| - |MinVCover(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Then $\boxed{S_G(U) = 2.|E_H(U)| - |U|}$ … Eqn (1)

- Proposition: If $(U, V \backslash U + w)$ is a <u>max cut</u> in G then U is a <u>vertex cover</u> in H.

Example 5: Max Cut

- Claim: $|MaxCut(G)| = 2.|E(H)| - |MinVCover(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Then $\boxed{S_G(U) = 2.|E_H(U)| - |U|}$ … Eqn (1)

- Proposition: If $(U, V \backslash U + w)$ is a <u>max cut</u> in G then U is a <u>vertex cover</u> in H.

  $\Rightarrow$ $S_G(U) = |MaxCut(G)| = 2.|E(H)| - |U|$

Example 5: Max Cut

- Claim: $|MaxCut(G)| = 2.|E(H)| - |MinVCover(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Then $\boxed{S_G(U) = 2.|E_H(U)| - |U|}$ … Eqn (1)

- Proposition: If $(U, V \backslash U + w)$ is a max cut in G then U is a vertex cover in H.

U must be a minVCover in H

$\Rightarrow S_G(U) = |MaxCut(G)| = 2.|E(H)| - |U|$

Example 5: Max Cut

- Claim: $|MaxCut(G)| = 2.|E(H)| - |MinVCover(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \backslash U + w)$ is a cut in G.

- Then $\boxed{S_G(U) = 2.|E_H(U)| - |U|}$ … Eqn (1)

- Proposition: If $(U, V \backslash U + w)$ is a max cut in G then U is a vertex cover in H.

  $\Rightarrow$ $S_G(U) = |MaxCut(G)| = 2.|E(H)| - |MinVCover(H)|$

Example 5: Max Cut

- Claim: $|\text{MaxCut}(G)| = 2.|E(H)| - |\text{MinVCover}(H)|$
- Proof: Let $V(H) = V$. Then $V(G) = V + w$.

  Suppose $(U, V \setminus U + w)$ is a cut in G.

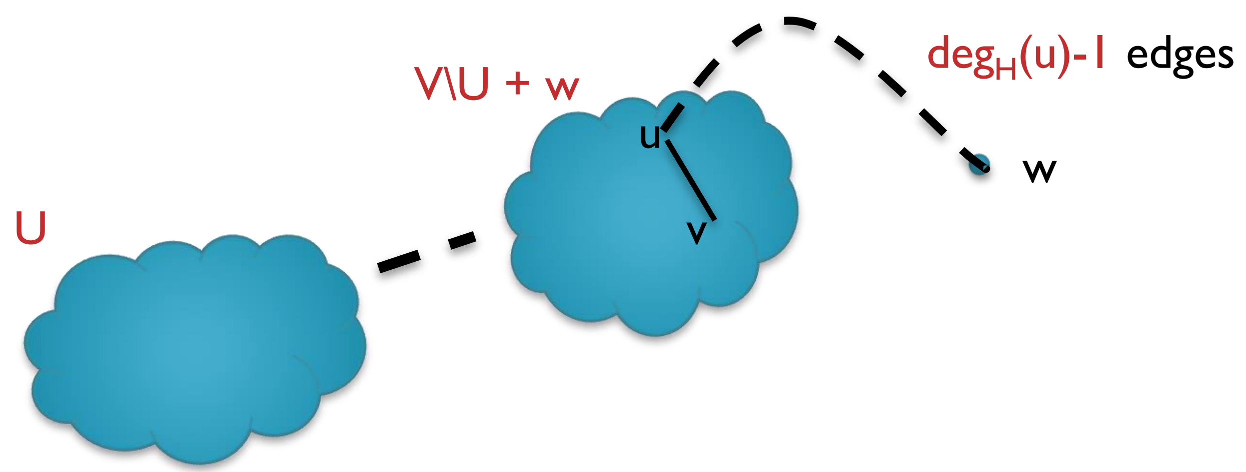
- Then $\boxed{S_G(U) = 2.|E_H(U)| - |U|}$ … Eqn (1)

- Proposition: If $(U, V \setminus U + w)$ is a max cut in G then U is a vertex cover in H.

Thus, the proof of the above claim follows from the proposition

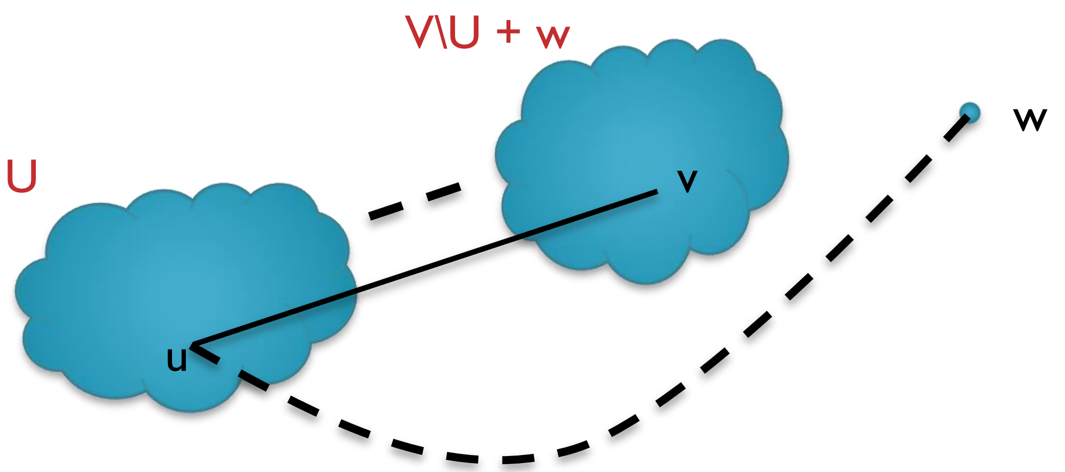
Example 5: Max Cut

- Proof of the Proposition: Suppose U is not a vertex cover



Example 5: Max Cut

- Proof of the Proposition: Suppose U is not a vertex cover



Gain: $deg_H(u)$-1 + 1 edges.
Loss: At most $deg_H(u)$-1 edges, these are the edges going from U to u.
Net gain: At least 1 edge. Hence the cut is not a max cut.