Lecture 8:
NP-Hard and NP-Completeness Reduction Problems

Winnie Zheng

November 2, 2023

1 Why is 3SAT NP-Hard?

Proof: We can demonstrate 3SAT is a NP-Hard problem by a reduction from Circuit Satisfiability (CSAT), since CSAT is NP-Hard:

$$CSAT \leq p3SAT$$

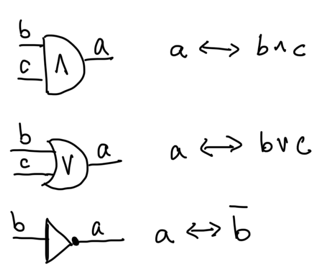
The following is an example of a CNF formula:

$$(x_1 \vee \bar{x_2} \vee x_3) \wedge (x_7 \vee \bar{x_9} \vee x_1) \wedge ...$$

Each clause is connected by an AND operator; each clause is composed of the OR result of 3 literals. Literals are variables or negated variables:

$$(x_1 \vee \bar{x_2} \vee x_3)$$

In Circuit SAT, the gates can be:



You can rewrite the gates expressions closer to 3SAT format:

1) $(a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee b) \wedge (\bar{a} \vee c)$

2) $(\bar{a} \vee b \vee c) \wedge (a \vee \bar{b}) \wedge (a \vee \bar{c})$

3) $(a \vee b) \wedge (\bar{a} \vee \bar{b})$

4) z

For a reduction, we must add an additional clause z to describe the output, as the output needs to be True. Since 3SAT requires every clause to have three literals, these boolean expressions can be useful as substitutions for clauses with fewer than 3 literals:

- a (a single literal) is equivalent to $(a \vee x \vee y) \wedge (a \vee \bar{x} \vee y) \wedge (a \vee x \vee \bar{y}) \wedge (a \vee \bar{x} \vee \bar{y})$ where x and y are new variables.
- $(a \vee b)$ (two literals) is equivalent to $(a \vee b \vee y) \wedge (a \vee b \vee \bar{y})$ where y is a new variable.

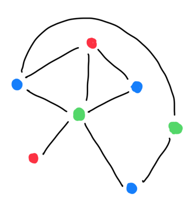
*Technicality: When transforming, make sure the problem doesn't blow up in size, which is true here.

Since we can reduce CSAT to 3SAT, then CSAT is NP-Hard.

2 Examples of NP-reduction

2.1 Graph Coloring

Given an undirected graph $G = (V, E)$, it is k-colored if $\exists$ a coloring map $C : V(G) \rightarrow 1, ..., k$ such that $\forall e = (v, u) \exists E : C(v) \neq C(u)$



The figure shows a 3-colorable graph.

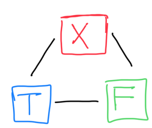
G is 2-colorable iff G is bipartite. This is easy to test in linear time. But 3-coloring is NP-hard!

Problem: Given G, is it 3-colorable?
Proof:

3-COLOR is NP-Hard

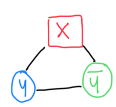
$$3SAT \leq_p 3COLOR$$

Gadget tools are used for this proof. We are building a graph that is hopefully 3-Colorable. So, three colors are needed for this triangle. We call the color used for vertex T, color "T". Same for "F". They will be associated with the True and False truth values:



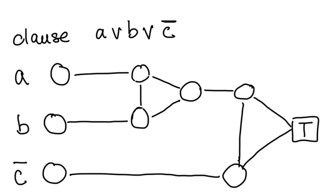
The blue T corresponds to T value, and green F corresponds to F value.

For each variable y we build, the variable gadget looks like the following diagram below. If y gets color T, and $\bar{y}$ is F, then y is True. If y gets color F and $\bar{y}$ is T, then y is False.

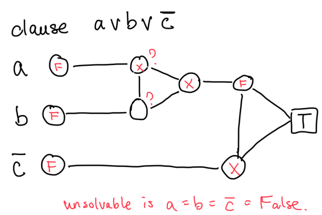


We can substitute T and F with variables.

We can use the variable gadgets to build a clause gadget. For example, for the clause gadget of the clause $(a \vee b \vee \vee \bar{c})$ look like this:



Claim: If $a \vee b \vee \bar{c}$ is False, then 3-coloring of the clause gadget is not possible:



If all three literals are False, the clause gadget won't be satisfied, which means

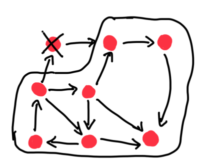
$$x_1 \vee x_2 \vee \bar{x_3} \text{ won't be true.}$$

If we can reduce to coloring the graph, then there is a satisfiable assignment with 3 valid colors, and if there is a satisfying assignment then the G is 3-colorable. This completes the reduction problem.

2.2 Hamiltonian Cycle

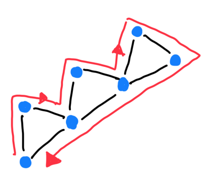
Definitions:

- Given a directed graph G, C is a Hamiltonian cycle in G iff C visits every vertex exactly once.
- G is Hamiltonian if it has a Hamiltonian cycle.



Example of a Hamiltonian G.

- C is an Eulerian cycle if it hits every edge exactly once (undirected).



Example of an Eulerian G.

Theorem: G is Eulerian iff G is 1) connected and 2) all degrees of edges are even.

2.2.1 Traveling Salesman Problem (TSP)

Given a complete and directed graph G, weights of edges s.t. $w : E \to \mathbb{R}$, and budget k, is there a cycle in G such that 1) visits every vertex exactly once $(v_1, v_2, v_3, v_4, ...v_n, v_1)$, 2) total cost c $\leq$ k?
Checking if the graph is Eulerian is easy. On the other hand, checking if the graph is Hamiltonian is NP-Hard.

$$c = w(v_1v_2) + w(v_2v_3) + ... + w(v_nv_1) \leq k$$

Claim:

TSP is NP-Hard.
HAM-CYCLE $\leq pTSP$

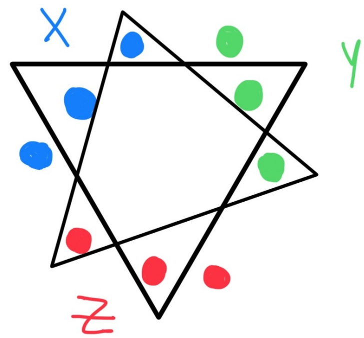
Proof: Given G(Ham-Cycle) = (V,E), a graph with a Hamiltonian cycle, we can assign the costs of each edge to 1, and every nonexisting edge to a cost of 2:

$$c = \begin{cases} 1, & e \in E(G) \\ 2, & e \notin E(G) \end{cases}$$

In this case, budget k = n(vertices), since G is Hamiltonian; a Hamiltonian cycle will have a cost of n. If it is not Hamiltonian, then any cycle must use at least one edge that does not exist on G (with a cost of 2), so it's cost will be greater than n. This completes the reduction. G is Hamiltonian iff the constructed TSP instance has a TSP cycle of cost $\leq$ n.

2.3 3D Matching (3DM)

There are 3 sets X,Y,Z where $|X| = |Y| = |Z| = n$ and a set of triplets $T \subset X \times Y \times Z$.
Question: Does there exist a set of triplets from T that are disjoint? Disjoint means that every element of X, Y, and Z is covered exactly once.



Can you finish this set of triplet so that T will be disjoint?

** Note: If this problem was for a double and not a triplet, it would be the classic bipartite perfect matching problem.

Theorem: 3DM is NP-Hard. (proof not covered in class).

2.4 Vertex Cover

G is an undirected graph. $X \subset V(G)$ is a **vertex cover** for every edge $e = \{v, w\} \in E(G)$, $e \wedge X \neq \emptyset$. In other words, X is a vertex cover if it touches every edge of G.

Opt: Find X with a min $|X|$.

Decision Version: Given G, k, does there exist vertex cover X such that $|X| \leq k$?

Claim:

IND (IndependentSet) $\leq$ p(VertexCover)
and therefore VertexCover is NP-Hard.

Proof: We are doing a reduction of the Independent Set problem to the Vertex Cover problem. Given graph G, X is a vertex cover iff V(G) \ X is independent due to the following Lemma:

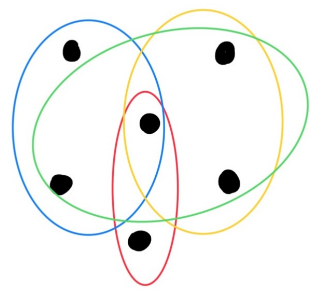
Lemma: A set S is a vertex cover in G iff V/S is an independent set in G.

2.5 SET COVER

Input: Given the input $S_1, S_2, S_3, ..., S_m$ of finite sets,

$$X = \bigcup_{i=1}^{m} S_i$$

Opt: Find the smallest subcollection of S whose union is X.
Decision: Given S_i, k, is there a family of k sets from S_i whose union is X?



possible covers: (blue, yellow, green, red), (blue, yellow, red), (blue, green, red)

Claim:

SET-COVER is NP-Hard.
because

$$VertexCover \leq pSET - COVER$$

Proof: The general idea is to start with a vertex cover graph, and list all the edges for each vertex. Create a set system. Ground set = all edges of G. Subsets = for every vertex v of G, the set of all edges of G that v is incidental.

2.6 General Rules for the next 2 Problems

Given a set X = $\{X_1, X_2, ..., X_n\}$ of positive integers. The size of the problem is about the number of bits needed to write it down.

$$size \approx \sum_{log_2} X_i$$

PARTITION: Given X, can we partition it into 2 subsets A, B such that

$$\sum_{x \in A} x = \sum_{x \in B} x$$

so that two groups have the same worth?

SUBSET SUM: Given X and a number S, is there a subset of X that sums up to a S?

$$\sum_{x \in X} x = S$$