NP-Completeness Reductions

3SAT $\leq_p$ Vertex Cover

3SAT – the language of satisfiable boolean formulas is 3CNF

Example:

$$(x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$$

Vertex Cover

Given a graph $G = (V, E)$ and an integer k, is there a subset V' of V with no more than k elements such that for every edge $e = (u, v) \in E$ either $u \in V'$ or $v \in V'$ or both.

VC $\in$ NP

- Given a V'
  - Iterate through the edges checking if one endpoint is in V'
  - check that $|V'| \leq k$

3SAT $\leq_p$ VC

- Given a boolean formula Φ in 3CNF we construct a graph G such that if Φ is satisfiable then G has a vertex cover of size $\leq k$
  if Φ is not satisfiable the G must not have a vertex cover of size k.

- let $U = \{x_1, x_2, \ldots, x_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ describe a boolean formula

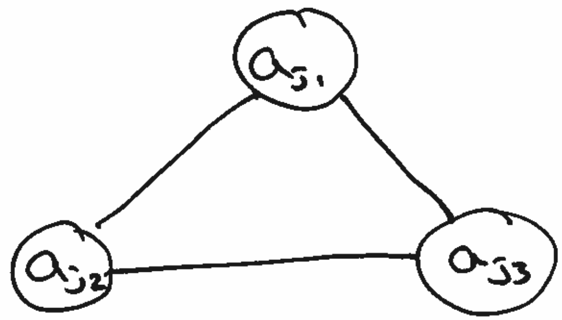
- construct a graph $G = (V, E)$ and a positive integer $k \leq |V|$ such that G has a vertex cover of size $\leq k$ iff all clauses in C are satisfiable.

- gadgets
  - truth-setting gadget
  - satisfaction-testing gadget

- For each $x_i \in U$ we include in G a gadget consisting of 2 nodes labelled x_i and $\overline{x_i}$ and a single edge between them

- For each clause $c_j \in C$ we include in G a gadget consisting of 3 nodes connected by edges labelled with their literal

$$C_j = (a_{j1} \vee a_{j2} \vee a_{j3})$$

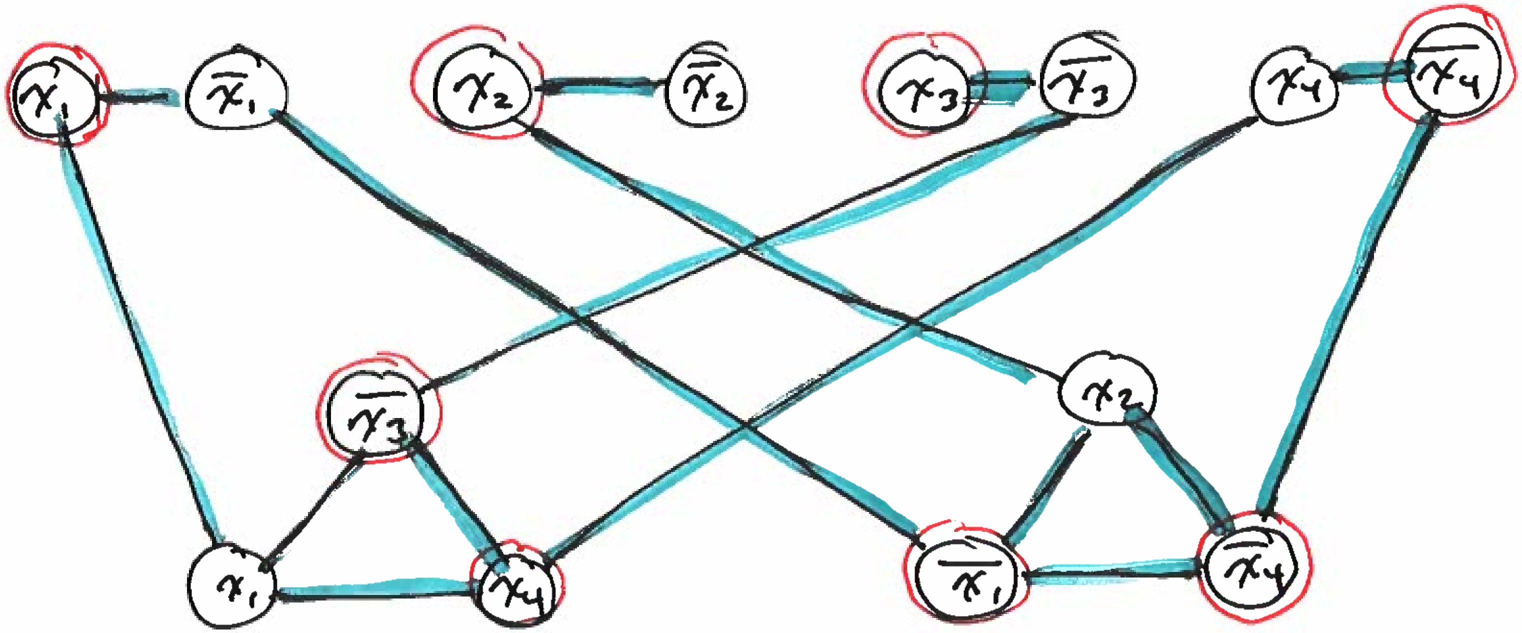


- For each clause $C_j \in C$ where $c_j = (l_1 \vee l_2 \vee l_3)$ include edges $E_{j1} = (a_{j1}, l_1)$, $E_{j2} = (a_{j2}, l_2)$ $E_{j3} = (a_{j3}, l_3)$

- Finally choose $K = n + 2M$

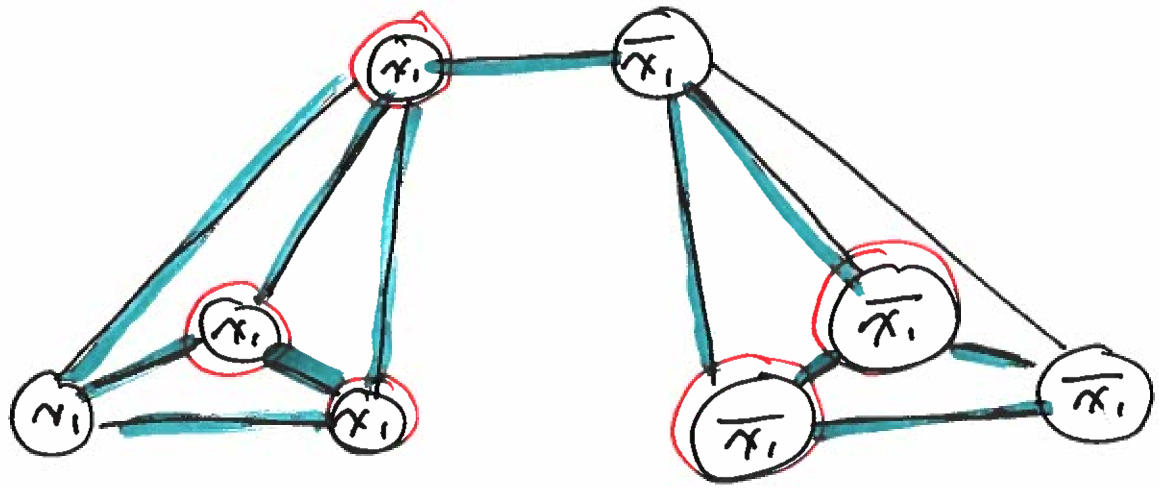
Example: $\Phi = (x_1 \vee \overline{x_3} \vee \underset{x_4}{\cancel{x_4}}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_4})$

$k = 4 + 2 \cdot 2 = 8$



$$\phi = (x_1 \vee x_1 \vee x_1) \wedge (\overline{x_1} \vee \overline{x_1} \vee \overline{x_1})$$

$k = 5$



To show the construction is valid

- suppose $V' \subseteq V$ is a vertex cover of G with $|V'| \leq k = n + 2m$
- V' must contain at least one vertex from each ~~th~~ truth-setting gadgets.
- V' must contain at least two vertices from each satisfaction-testing gadget.
- ~~sinel~~ since this gives a total of k vertices already, we can conclude V' must contain exactly one vertex from each truth-setting gadget and exactly two from each satisfaction testing gadget.

- The choice of vertex in each truth-setting gadget in V' induces a truth assignment

  set x_i = True if $x_i \in V'$ and x_i = False if $\overline{x_i} \in V'$

- This truth assignment satisfies each clause in C
- As observed exactly two vertices a_{j1}, a_{j2}, a_{j3} are in V' so only two of the three edge E_{jk} can be covered by these vertices
- The third one must be covered by the vertex l_i. This means $l_i \in V'$ i.e. l_i evaluates to True hence C_j is satisfied.

Conversely

- suppose we have a satisfying assignment for each clause in C
- let V' include each x_i such that $x_i = \text{True}$ and each $\overline{x_i}$ such that $x_i = \text{False}$
- Since each C_j is satisfied, this means at least one of the three E_{jk} is covered by vertex l_i
- To cover the other two edges E_{jk} as well as the triangle we can pick any two vertices a_{jk}
- The overall size of V' is $2m + n$
- so V' is a vertex cover of size k

What to do when you are asked to solve an NP-Complete problem?

- Give up
- Approximate (sub-optimal solutions)
- solve an easier subset
- Only solve small inputs