The Vertex Cover Problem

Shangqi Wu

Presentation of CS 525

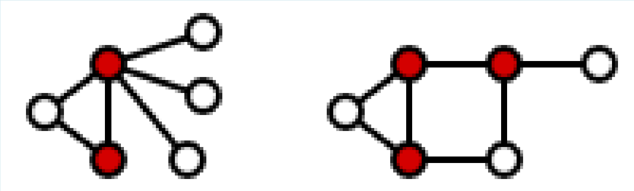
March 11th, 2016

Vertex-Cover Problem Definition

If G is an undirected graph, a vertex cover of G is a subset of nodes where every edge of G touches one of those nodes. The vertex cover problem asks whether a graph contains a vertex cover of a specified size:

$$Vertex - Cover = \{< G, k >|$$

$$G\ is\ an\ undirected\ graph\ that\ has\ a\ k - node\ vertex\ cover\}$$



Definition of NP-Complete

- Definition 7.34
  - A language B is NP-Complete if it satisfies the following conditions:
    - B is in NP.
    - Every A in NP can be polynomial-time reducible to B.
- Theorem 7.36
  - Given that A is NP-complete and B is in NP, if $A \leq_p B$, then B is NP-complete.

Reduction & Reducibility

- Definition 7.28
    - A function of $f\colon \sum * \rightarrow \sum *$ is polynomial time computable function if some polynomial time Turing Machine M exists that halts with just $f(\omega)$ on its tape, when started on any input ω.
- Definition 7.29
    - Language A is polynomial time mapping reducible, or simply polynomial time reducible, to language B, written $A \leq_p B$, if a polynomial time computable function $f\colon \sum * \rightarrow \sum *$ exists, where for every ω,

$$\omega \in A \Leftrightarrow f(\omega) \in B$$

The function f is called the polynomial time reduction of A to B.

Proof Idea

- To show Vertex-Cover problem is NP-complete, there are 2 steps:
  - Show Vertex-Cover is in NP.
  - Show all NP-problems are polynomial time reducible to Vertex-Cover problem.
    - Choose another NP-complete problem, e.g., 3-SAT, to show that it is polynomial reducible to Vertex-Cover.

Step 1: Vertex-Cover in NP

- Regard vertex cover of k nodes as a certificate to verify the Vertex-Cover solution. The verifier V for Vertex-Cover is constructed as description below.

V = On input <<G, k>, c>:

Test if c is a subgraph containing k nodes in graph G.
Test if c covers all edges in graph G.
If the answers are both yes, the verifier accepts, otherwise rejects.

Step 2: Reduction from 3-SAT

- The reduction converts a 3NF formula ϕ in 3-SAT into a graph G and a number k, and ϕ is satisfiable when G has a vertex cover with k nodes.
- Graph G should contain gadgets to imitate behavior of variable and clauses in 3-SAT, and the structure of gadgets are constructed as description below.
  - Variable gadget:
    - For each variable x in ϕ, produce two nodes x and $\bar{x}$, and connect them with an edge.
  - Clause gadget:
    - For each clause in ϕ, there are 3 nodes which are labeled by literals of the clause, then 3 nodes are connected to each other and to the nodes in variable gadgets having the same labels.

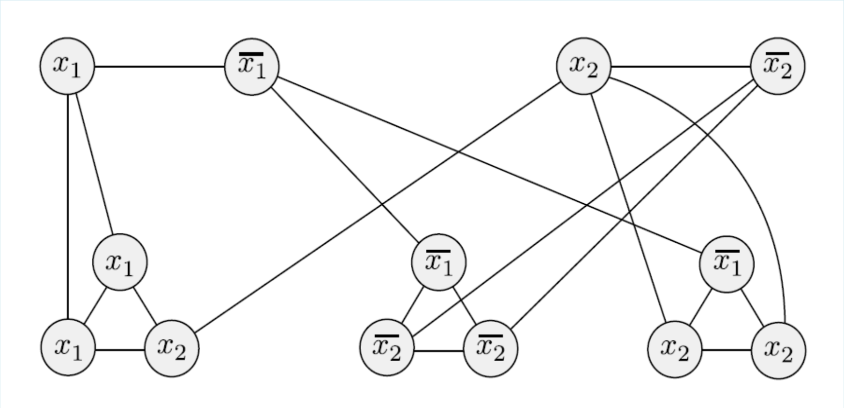
Reduction Continued

- Let m and l be the number of variables and clauses respectively. Then we have $2m + 3l$ nodes in G. Let k be $k = m + 2l$.
- Here provides an example of reduction:

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

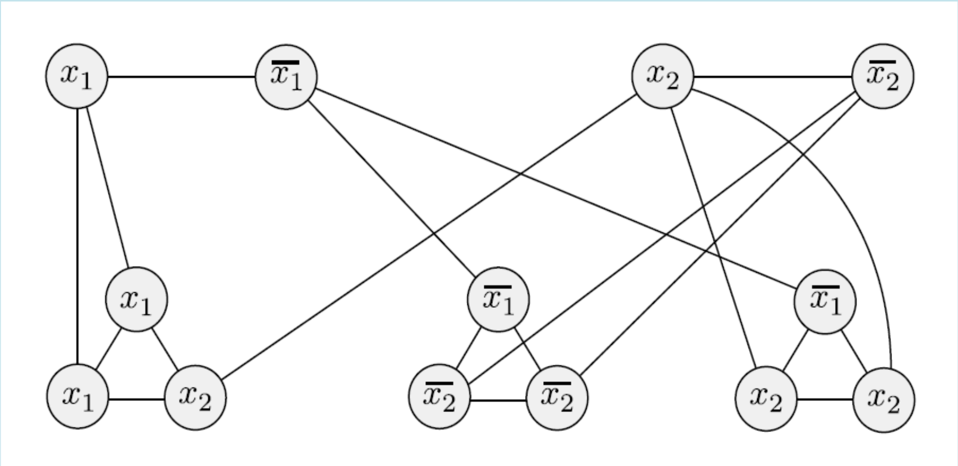
Reduction produces <G, k> from ϕ, where k = 8.

- G is presented in following figure:



Reduction Continued

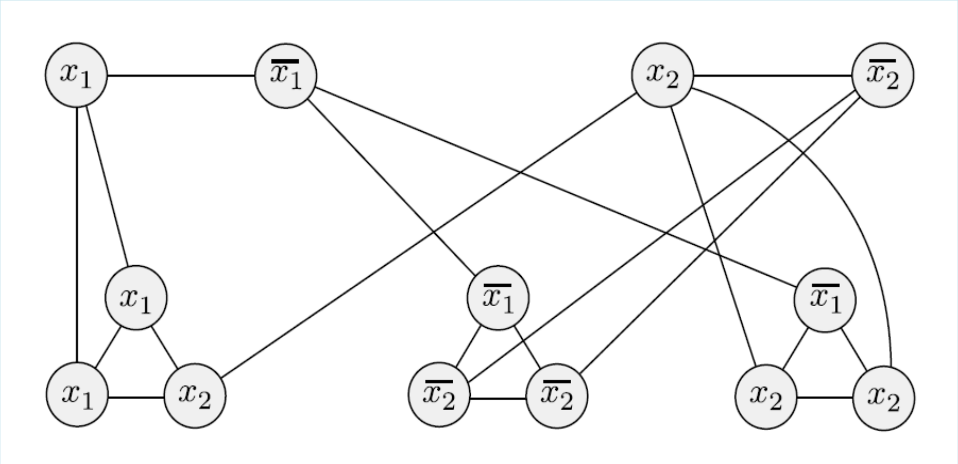
- This step shows if ϕ satisfiable, G has a vertex cover of size k.
  - For each variable gadget, take the nodes which are corresponding to the true literal in the assignment into the vertex cover.
  - For each clause gadget, select one true literal and put rest 2 nodes into the vertex cover.
  - Vertex cover contains k nodes, and it covers all edges in graph G.



$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

Reduction Continued

- This step shows if G has a vertex cover of size k, ϕ is satisfiable.
- The vertex cover must contain following to cover correct edges:
  - One node in each variable gadget.
  - 2 nodes in every clause gadget.
- We take nodes of variable gadgets that are in vertex cover and assign true to corresponding literals.
- Each of 3 edges connecting the variable gadgets with each clause gadget is covered, and only 2 nodes of the clause gadget are in the vertex cover.



$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

Conclusion

- Based on proof above, we can conclude that ϕ is satisfiable if and only if G has a vertex cover with k nodes.
- Since Vertex-Cover problem is in NP, and it is reducible to 3-SAT problem, which is shown to be NP-complete before, therefore Vertex-Cover is NP-complete.

Thank you!