The NP-completeness of Vertex Cover

Pilu Crescenzi and Viggo Kann

University of Florence and KTH

October 2011
Basic definitions

- **Class NP**
  - Set of decision problems that admit “short” and efficiently verifiable solutions
  - Formally, \( L \in \text{NP} \) if and only if there exist
    - polynomial \( p \)
    - polynomial-time machine \( V \)
    - such that, for any \( x \),
      \[
      x \in L \iff \exists y (|y| \leq p(|x|) \land V(x, y) = 1)
      \]

- **Polynomial-time reducibility**
  - \( L_1 \leq L_2 \) if there exists polynomial-time computable function \( f \) such that, for any \( x \),
    \[
    x \in L_1 \iff f(x) \in L_2
    \]

- **NP-complete problem**
  - \( L \in \text{NP} \) is NP-complete if any language in \( \text{NP} \) is polynomial-time reducible to \( L \)
    - Hardest problem in \( \text{NP} \)
Basic results

Cook-Levin theorem

Sat problem

Given a boolean formula in conjunctive normal form (disjunction of conjunctions), is the formula satisfiable?

Sat is NP-complete

3-Sat

Each clause contains exactly three literals

3-Sat is NP-complete

Simple proof by local substitution

\[ l_1 \Rightarrow (l_1 \lor y \lor z) \land (l_1 \lor y \lor \overline{z}) \land (l_1 \lor \overline{y} \lor z) \land (l_1 \lor \overline{y} \lor \overline{z}) \]

\[ l_1 \lor l_2 \Rightarrow (l_1 \lor l_2 \lor y) \land (l_1 \lor l_2 \lor \overline{y}) \]

\[ l_1 \lor l_2 \lor l_3 \Rightarrow l_1 \lor l_2 \lor l_3 \]

\[ l_1 \lor l_2 \lor \cdots \lor l_k \Rightarrow \]

\[ (l_1 \lor l_2 \lor y_1) \land (\overline{y_1} \lor l_3 \lor y_2) \land (\overline{y_2} \lor l_4 \lor y_3) \land \cdots \land (\overline{y_{k-3}} \lor l_{k-1} \lor l_k) \]
Problem definition: Vertex Cover

Given a graph $G = (N, E)$ and an integer $k$, does there exist a subset $S$ of at most $k$ vertices in $N$ such that each edge in $E$ is touched by at least one vertex in $S$?

- No polynomial-time algorithm is known
- Is in NP (short and verifiable solution):
  - If a graph is “$k$-coverable”, there exists $k$-subset $S \subseteq N$ such that each edge is touched by at least one of its vertices
  - Length of $S$ encoding is polynomial in length of $G$ encoding
  - There exists a polynomial-time algorithm that verifies whether $S$ is a valid $k$-cover
    - Verify that $|S| \leq k$
    - Verify that, for any $(u, v) \in E$, either $u \in S$ or $v \in S$
Reduction of 3-Sat to Vertex Cover:

Technique: component design

- For each variable a gadget (that is, a sub-graph) representing its truth value
- For each clause a gadget representing the fact that one of its literals is true
- Edges connecting the two kinds of gadget

Gadget for variable $u$:

$\begin{array}{c}
F_u \\
\end{array}$

One vertex is sufficient and necessary to cover the edge

Gadget for clause $c$:

$\begin{array}{c}
F_c \\
S_c & T_c
\end{array}$

Two vertices are sufficient and necessary to cover the three edges

$k = n + 2m$, where $n$ is number of variables and $m$ is number of clauses
Connections between variable and clause gadgets

- First (second, third) vertex of clause gadget connected to vertex corresponding to first (second, third) literal of clause
- Example: \((x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor x_3) \land (x_1 \lor x_2 \lor \overline{x_3})\)

Idea: if first (second, third) literal of clause is true (taken), then first (second, third) vertex of clause gadget has not to be taken in order to cover the edges between the gadgets.
Proof of correctness

- Show that Formula satisfiable $\Rightarrow$ Vertex cover exists:
  - Include in $S$ all vertices corresponding to true literals
  - For each clause, include in $S$ all vertices of its gadget but the one corresponding to its first true literal
  - Example
    - $(x_1 \lor x_2 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor \overline{x_3})$
    - $x_1$ true, $x_2$ and $x_3$ false

- Show that Vertex cover exists $\Rightarrow$ Formula satisfiable:
  - Assign value true to variables whose $p$-vertices are in $S$
  - Since $k = n + 2m$, for each clause at least one edge connecting its gadget to the variable gadgets is covered by a variable vertex
    - Clause is satisfied