Tema 4 - Reducere k-Vertex-Cover $\leq_P$ SAT

Nume, Prenume: Blaga Ana-Maria-Andreea

Grupa: 324CB

Nume utilizator HackerRank: blagaandreea238

Input for k Vertex Cover

Graph G = (V, E), where:

V = {1, 2, …, n} is the set of vertices in the graph.

E = $\{e_1, e_2, \ldots, e_m\}$ is the set of edges in the graph, where $e_i = \{(i, j) \mid i, j \in V\}$.

K is the number of vertices wanted in the Vertex Cover.

Encoding k Vertex Cover in SAT

Such a cover has **k** slots (where **k** is the number of vertices wanted in the cover).

$$\mathbf{S_1 \quad S_2 \quad S_3 \quad \ldots \quad S_k}$$

We invent **k * n** variables (k variables for each vertex in the graph):

$$\mathbf{X_{i,\,j}} \text{ where vertex } \mathbf{i} \text{ occupies slot } \mathbf{S_j},\ \mathbf{i \in [1, n]} \text{ and } \mathbf{j \in [1, k]}$$

$$\mathbf{X_{i,\,j} = (i - 1) * k + j}$$

For example, **vertex i** has the following **k variables** in the SAT encoding: $\mathbf{X_{i,1}, X_{i,2} \ldots X_{i,k}}$.

Encoding rules for the Transformation:

1. Every edge is incident to at least one vertex in the Vertex Cover.

For each **(i, j) ∈ E** the following clause is created: $\mathbf{(X_{i,1} \vee X_{j,1} \vee X_{i,2} \vee X_{j,2} \vee \ldots \vee X_{i,k} \vee X_{j,k})}$.

The clauses created at this point verify the existence of the k Vertex Cover.

A clause is created for each edge in the graph consisting of all its defining vertices variables, interconnected by ‘V’.

For an edge to be covered by the Vertex Cover at least one of its ends (vertices) must be in the Vertex Cover. ⇔ At least one of the variables corresponding to its defining vertices (ends) must be set as True.

If at least one edge is not covered then the respective combination of vertices cannot be a Vertex Cover. ⇔ If there is any clause created at this step evaluated to False (no variable in it is set to True) then it means that an edge was not covered.

|E| = m clauses

2. Each slot $\mathbf{S_j}$ must be occupied.

For each slot **Sj (j ∈ [1, k])** the following clause is created: $\mathbf{(X_{1,\,j} \lor X_{2,\,j} \lor \ldots \lor X_{k,\,j})}$.

The clauses created at this point verify that there is no empty slot left in the k Vertex Cover. ⇔ At least one of the variables for each slot must be set to True (when creating the variables for a vertex using the previously explained formula, the variable j is defining the slot, that's why j ∈ [1, k] and why we create k variables for each vertex).

A clause is created for every slot consisting of the vertices' variables for the current slot, interconnected by 'V'.

k clauses

3. No vertex can appear twice in the Vertex Cover.

For each **vertex i** and each pair of slots $\mathbf{S_a \neq S_b}$ **(a, b € [1, k], a < b)** the following clause is created: $\mathbf{(\sim X_{i,\,a} \lor \sim X_{i,\,b})}$ where **i ∈ [1, n].**

The clauses created at this point verify that a vertex does not appear both in positions a and b of the Vertex Cover. ⇔ Only one of each vertex's corresponding variables can be set as True.

If the K variables set as True do not correspond to one different vertex in the Vertex Cover, then there is at least one edge left outside the Vertex Cover. So, each one of the k variables set as True must correspond to one different vertex in the K Vertex Cover.

For each vertex in the graph clauses consisting of the negated forms of any two of its variables are created.

$O(n^3)$ clauses

4. No two vertices **i** and **j** can occupy the same slot $\mathbf{S_a}$.

For each slot $\mathbf{S_a}$ **(a ∈ [1, k])** and each pair of vertices **i ≠ j (i, j ∈ [1, n], i < j)** the following clause is created: $\mathbf{(\sim X_{i,\,a} \lor \sim X_{j,\,a})}$.

The clauses created at this point verify that only one vertex appears on the a'th position of the Vertex cover. ⇔ Only one variable for the slot S_a can be set as True. So, each slot can be occupied by one and only variable.

For each slot clauses consisting of the negated forms of any two of the vertices' variables for it are created.

$O(n^3)$ clauses

Number of clauses in the reduction: $O(n^3)$.

$|E| + k + O(n^3) + O(n^3) \Rightarrow O(n^3)$

$|E| = m \leq n * (n - 1) / 2 \Rightarrow O(n^2)$

Maximum number of edges in a graph = $n * (n - 1) / 2$

$k \leq n \Rightarrow O(n)$

$$G = (V, E), k \Rightarrow T \Rightarrow \varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$$

There exists a **Vertex Cover** of size **k** in the graph iff (if and only if) **φ** is satisfiable.

(a) Number of clauses in the CNF formula (**φ**): **$O(n^3)$.**
(b) The execution time is proportional with the number of clauses in the CNF formula.

(a) + (b) => The execution time **∈ $O(n^3)$** (polynomial time) => **T is** a **polynomial** transformation.