

5MC10: Combinatorial Algorithms

NP-complete problems

AmirHossein Ghamarian

December 2, 2008

1. The **CLIQUE** problem is the decision problem defined in your book: Given an undirected graph $G = (V, E)$ and a natural number k , decide whether there is a clique of size k in G . The **MAXIMUM-CLIQUE** problem is an optimization problem defined by: Given an undirected graph $G = (V, E)$, determine a set of vertices $U \subseteq V$ which form a clique of maximum size in G . The **MAXIMUM-CLIQUE-SIZE** problem is another optimization problem defined as follows: Given an undirected graph $G = (V, E)$, determine the size of a maximum clique in G .

Prove the following sequence of polynomial reductions:

$$\text{CLIQUE} \leq_P \text{MAXIMUM-CLIQUE} \leq_P \text{MAXIMUM-CLIQUE-SIZE} \leq_P \text{CLIQUE}.$$

Solution: $\text{CLIQUE} \leq_P \text{MAXIMUM-CLIQUE}$ is easy to verify because if we know the maximum clique then we know the maximum clique size. Then we only need to check whether it is larger than k . To show the second part, $\text{MAXIMUM-CLIQUE} \leq_P \text{MAXIMUM-CLIQUE-SIZE}$, we need to show that by knowing the maximum clique size k , we can find the vertices which are part of the clique. Since the number of different k vertices out of the total vertices is a polynomial number, and verifying whether each k chosen vertices construct a clique is also polynomial, therefore, we can polynomially check all the k vertices of the graph. To show the last part $\text{MAXIMUM-CLIQUE-SIZE} \leq_P \text{CLIQUE}$ we need to prove that if we know whether the graph has the clique of size at least k , then we can polynomially find the maximum clique size of the graph. To show this we can start with the maximum number which is the number of vertices of the graph, as the k parameter of the clique, and decrease k one by one and stop as soon as **CLIQUE** returns yes, and k will be the maximum clique size of the graph.

2. It is shown that **VERTEX-COVER** is NP-hard by reduction $\text{CLIQUE} \leq_P \text{VERTEX-COVER}$. Prove the converse reduction, namely $\text{VERTEX-COVER} \leq_P \text{CLIQUE}$.

Solution: Let $\langle G, k \rangle$ be an instance of VERTEXCOVER. We polynomially transform G into an instance $\langle G', k' \rangle$ of CLIQUE such that G has a vertex cover of size $\leq k$ iff G' has a clique of size $\geq k'$. If $G = (V, E)$ then the desired $G' = (V', E')$ is defined by:
 $V' = V$ and $E' = (V \times V) - E$.

There is no loss of generality in assuming that $k \leq n = |V|$. Set $k' = n - k$. The transformation from $\langle G, k \rangle$ to $\langle G', k' \rangle$ is polynomial time. Consider now a vertex cover $U \subseteq V$ for G of size $\leq k$ and let $T = V - U$. For all $v, w \in V$ if $(v, w) \in E$ then $v \in U$ or $w \in U$ or both (because U is a vertex cover). The contrapositive of this implication is: For all $v, w \in V$ if $v \notin U$ and $w \notin U$ then $(v, w) \notin E$. In other words, $V - U = T$ is a clique for G of size $\geq n - k = k'$.

Conversely, if T is a clique for G' of size $\geq k'$, the set $U = V - T$ is a vertex cover for G of size $\leq k$.

3. Show that hamiltonian-path problem can be solved in polynomial time on directed acyclic graphs. Give an efficient algorithm for the problem.

Solution: Since the graph is acyclic and directed, we can apply the topologically sort it. Then verifying the existence of a Hamiltonian path is verifying whether all the adjacent vertices are connected in the topologically sorted graph.

4. The **longest-simple-cycle problem** is the problem of determining a simple cycle (no repeated vertices) of maximum length in a graph. Show that this problem is NP-Complete.

Solution: Verifying the fact that the problem belongs to NP is trivial. We use Hamiltonian cycle problem for the reduction. In fact, the Hamiltonian cycle problem is the special case of the lsc problem. Iff the longest simple cycle of the graph is as large as the number of vertices of the graph, then the graph has a hamiltonian cycle.

5. Given an integer $m \times n$ matrix A and an integer m -vector b , the **0-1 integer programming problem** asks whether there is an integer n -vector x with elements in the set $\{0, 1\}$ such that $Ax \geq b$. Prove that 0 – 1 integer programming is NP-complete. (Hint: Reduce from 3-SAT)

Solution: The complete solution can be found in

<http://www.cs.uky.edu/~lewis/cs-heuristic/text/class/more-np.html>