Lab11-NP Reduction

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2020.

∗ If there is any problem, please contact TA Shuodian Yu.

∗ Name: Futao Wei Student ID: 518021910750 Email: weifutao@sjtu.edu.cn

1. What is the "certificate" and "certifier" for the following problems?

   (a) *ZERO-ONE INTEGER PROGRAMMING*: Given an integer $m \times n$ matrix A and an integer m-vector b, is there an integer n-vector x with elements in the set $\{0, 1\}$ such that $Ax \leq b$.

   (b) *SET PACKING*: Given a finite set U, a positive integer k and several subsets $U_1, U_2, \ldots, U_m$ of U, is there k or more subsets which are disjoint with each other?

   (c) *STEINER TREE IN GRAPHS*: Given a graph $G = (V, E)$, a weight $w(e) \in Z_0^+$ for each $e \in E$, a subset $R \subset V$, and a positive integer bound B, is there a subtree of G that includes all the vertices of R and such that the sum of the weights of the edges in the subtree is no more than B.

   Solution.

   (a) **Certificate:** An integer n-vector x with elements in the set $\{0, 1\}$.
   Certifier: An algorithm that checks if $Ax \leq b$.

   (b) **Certificate:** A collection C of subsets of U whose size is no less than k.
   Certifier: An algorithm that checks if the elements in C are disjoint with each other.

   (c) **Certificate:** A subtree of G that includes all the vertices of R.
   Certifier: An algorithm that checks if the sum of the weights of the edges in the subtree is no more than B.

   □

2. Algorithm class is a democratic class. Denote class as a finite set S containing every students. Now students decided to raise a student union $S' \subseteq S$ with $|S'| \leq K$.

   As for the members of the union, there are many different opinions. An opinion is a set $S_o \subseteq S$. Note that number of opinions has nothing to do with number of students.

   The question is whether there exists such student union $S' \subseteq S$ with $|S'| \leq K$, that S' contains at least one element from each opinion. We call this problem *ELECTION* problem, prove that it is NP-complete.

   Proof. First, *ELECTION* problem is NP, since we can find a poly-time certifier. What the certifier does is to check the students in S' one by one to determine whether there's at least one element from each opinion group.
   Next, we intend to establish NP-completeness of *ELECTION* problem by proving *SET COVER* $\leq_P$ *ELECTION*.
   Instance of *SC*: A set U; a collection of m subsets of U $(S_1, S_2, \ldots, S_m)$; an integer K
   Instance of *ELECTION*: The same set U, whose elements denote all kinds of opinions; the same collection of m subsets of U $(S_1, S_2, \ldots, S_m)$, S_i denotes the opinions the i^{th} student holds; the same integer K; a set S of all the students
   (Note: The instance of *ELECTION* above can be transformed to the original form)
   Then we prove that there exists a set cover of size $\leq K$ iff there exists a student union $S' \subseteq S$

with $|S'| \leq K$.

$\Longrightarrow$:

If there exists a set cover of size $\leq K$, namely, a collection $S_{i_1}, S_{i_2}, \ldots, S_{i_n}$, $n \leq K$, then we can choose those students numbered $i_1, i_2, \ldots, i_n$ to form a student union, whose opinions will definitely cover all the opinions.

$\Longleftarrow$:

If there exists a student union $S' \subseteq S$ with $|S'| \leq K$, suppose their numbers are $i_1, i_2, \ldots, i_n$, $n \leq K$, then the collection $S_{i_1}, S_{i_2}, \ldots, S_{i_n}$ forms a set cover.

Hence *SET COVER* $\leq_P$ *ELECTION*. *ELECTION* problem is NP-complete. $\square$

3. Not-All-Equal Satisfiability (NAE-SAT) is an extension of SAT where every clause has at least one true literal and at least one false one. NAE-3-SAT is the special case where each clause has exactly 3 literals. Prove that NAE-3-SAT is NP-complete. (Hint : reduce 3-SAT to NAE-k-SAT for some $k > 3$ at first)

   Proof. First, NAE-3-SAT is NP, since we can find a poly-time certifier. What the certifier does is to check whether every clause has at least one true literal and at least one false one. Next, we'll establish its NP-completeness by proving 3-SAT $\leq_P$ NAE-3-SAT. However, we'll prove 3-SAT $\leq_P$ NAE-4-SAT first as an intermediary.

   Instance of 3-SAT: literals $x_1, x_2, \cdots, x_n$; clauses, for example, $(x_i \vee x_j \vee x_k)$

   Instance of NAE-4-SAT: literals $z, y_1, y_2, \cdots, y_n$, which follow the rule that $y_i = \bar{z}$ or $y_i = z$ if x_i is true or false respectively; clauses, (z, y_i, y_j, y_k) corresponding to $(x_i \vee x_j \vee x_k)$, similar for negations

   Now we need to prove that the instance of 3-SAT is satisfiable iff the instance of NAE-4-SAT is satisfiable.

   $\Longrightarrow$:

   If there exists a truth assignment for the instance of 3-SAT, then in each clause there has to be at least one literal x_i (or its negation $\bar{x_i}$) that is true. According to the mapping rule above, there has to be at least one literal y_i (or its negation $\bar{y_i}$) that is equal to $\bar{z}$ in each clause, i.e., every clause is true and the instance of NAE-4-SAT is satisfiable.

   $\Longleftarrow$:

   If there exists a truth assignment for the instance of NAE-4-SAT, then in each clause there has to be at least one literal y_i (or its negation $\bar{y_i}$) that is equal to $\bar{z}$. According to the mapping rule above, there has to be at least one literal x_i (or its negation $\bar{x_i}$) that is true in each clause, i.e., every clause is true and the instance of 3-SAT is satisfiable.

   Hence we've shown that 3-SAT $\leq_P$ NAE-4-SAT.

   To reduce NAE-4-SAT to NAE-3-SAT, we introduce a variable w to break each 4-variable clause into two 3-variable ones, i.e., $(z, y_i, y_j, y_k) = (y_i, y_j, w) \wedge (\bar{w}, y_k, z)$. If $y_i = y_j = y_k = z$, both sides are false. If not, we can set the value of w so that both sides are true.

   Therefore, 3-SAT $\leq_P$ NAE-3-SAT. NAE-3-SAT is NP-complete. $\square$

4. In the Lab10, we have introduced Minimum Constraint Data Retrieval Problem (MCDR). Prove that MCDR (Version 1 or 2) is NP-complete. (Hint : reduce from VERTEX-COVER or 3-SAT)

   MCDR problem:

   Input:

   - Data item set $D = \{d_1, d_2, \cdots, d_k\}$, each d_i with length l_i as time units
   - Broadcasting channel set $\mathbf{C} = \{C_1, C_2, \cdots, C_n\}$, each C_i with data sequence S_i to be repeatedly broadcasted

- Broadcasting starting time t_i for each C_i, which is defined to obtain the time differences between two channels
- The data item set $D_q \subseteq D$, which is needed by the client

Output (version 1):

$$ans = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{if } P \text{ is false} \end{cases}$$

where P: There exists a legal sequence of (C_k, d_k) which achieves a switch cost less than h_0 (some specific value).

Proof. First, MCDR is NP since we can find a poly-time certifier, which checks whether the switch cost of the sequence is less than h_0.
Next, we'll establish its NP-completeness by proving that VERTEX-COVER $\leq_P$ MCDR.
Instance of VERTEX-COVER: A graph $G = (V, E)$; an integer k
Instance of MCDR: A data item set $D_q \subseteq D$, which is needed by the client, each element corresponds to an edge in E; a broadcasting channel set $\mathbf{C} = \{C_1, C_2, \cdots, C_n\}$, each C_i corresponds to a vertex in V (channel C_i has data d_j if C_i's corresponding vertex is at an end of d_j's corresponding edge; the sequence of data broadcast in C_i is arbitrary); an integer $h = k$
We need to prove that there exists a vertex-cover of size $\leq k$ iff there exists a channel sequence of size $\leq h$ that covers the data in D_q.
$\Longrightarrow$:
If there exists a vertex-cover of size $\leq k$, denoted as S, then we can choose the corresponding channels in $\mathbf{C}$ to form a sequence of size $\leq h$ that covers the data in D_q.
$\Longleftarrow$:
If there exists a channel sequence of size $\leq h$ that covers the data in D_q, then we can choose the corresponding vertices as a vertex-cover.
Hence we've reduced VERTEX-COVER to MCDR. We can conclude that MCDR is NP-complete. □

Remark: Please include your .pdf, .tex files for uploading with standard file names.