Lab11-NP Reduction

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

* If there is any problem, please contact TA Shuodian Yu.
* Name:Yulong Hui   Student ID:518030910059   Email: qinchuanhuiyulong@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 $U_1, U_2, \dots, U_m$ 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. Check that n-vector x satisfies: $Ax \leq b$.

   (b) **Certificate.** A collection set C of subsets $U_1, U_2, ..., U_m$, whose cardinality is not less than k.
   Certifier. Check that the sets in C are disjoint with each other.

   (c) **Certificate.** A subtree of G that includes all the vertices of R.
   Certifier. Check that 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. It is known to us that VERTEX-COVER problem is NP-complete, so we can just finish this proof by proving: VERTEX-COVER $\leq_P$ ELECTION.

   Assume any graph $G = (V, E)$, and let S have the same meaning as V. Then students in S equal to the vertices in V. Let every edge e_i (contains 2 vertices) in V equals to an opinion S_i, which means there are always 2 students in every opinion. Therefore, if we need to solve VERTEX-COVER problem and find a subset of V which have at least one element in every edge e_i, we can transform it to a special ELECTION problem and find a student union $S' \subseteq S$ which contains at least one element in each opinion S_i.

   Then we has proved: VERTEX-COVER $\leq_P$ ELECTION, so ELECTION problem is NP-complete.

   □

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, we can reduce the 3-SAT problem to a NAE-4-SAT problem:

Assume one clause as $(x_1 \bigvee x_2 \bigvee x_3)$, then it equals to a NAE-4 clause $N_4(x_1, x_2, x_3, F)$. In the NAE-4 clause, the result will be true iff it has at least one true value among x_1, x_2, x_3, which can solve the 3-SAT prblem. Therefore, we can get: 3-SAT $\leq_P$ NAE-4-SAT.

Then we can reduce the NAE-4-SAT problem to a NAE-3-SAT problem:

Assume one clause as $N_4(x_1, x_2, x_3, x_4)$, and construct a NAE-3 format as follows: (s is a binary variable)

$$N = N_3(x_1, x_2, s) \bigwedge N_3(x_3, x_4, \overline{s})$$

It's easy to see that if x_1, x_2, x_3, x_4 are all equal, no matter what value the s is, N is always false. Then if x_1, x_2, x_3, x_4 are not all equal, we can always find a value for s and make N true. Therefore, we can sovle the NAE-4-SAT problem by constructing a such NAE-3 format formula. Then, we can get: NAE-4-SAT $\leq_P$ NAE-3-SAT.

So, 3-SAT $\leq_P$ NAE-4-SAT $\leq_P$ NAE-3-SAT, and NAE-3-SAT is NP-complete.

□

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)

Proof. We should make some definistions clear:

MCDR-V1 means: Does there exist a scenario whose switch-number $\leq k$, while access-latency is no more than t.

MCDR-V2 means: Does there exist a scenario whose access-latency $\leq k$, while switch-number is no more than h.

At first, we will prove MCDR-V1 by reducing from VERTEX-COVER:

Assume a graph $G = (V, E)$, if there is a vertex v_i corresponding to several edges $e_1, e_2, ..e_j$, then we can construct a channel c_i with continuous data-items $d_1, d_2, ...d_j$ in it. That's to say a channel equals to a vertex, and a data-item equals to a edge.

Then we can transform the VERTEX-COVER problem (is there a vertex set of size $\leq k$ which can cover all the edges) to a MCDR-V1 problem (is there a scenario of switch-time $\leq (k-1)$ which can cover all the data-items). Specially, in this case the constraint t of access-time is infinite, and we just need to find the smallest switch-time which equals to "the size of vertex-set - 1".

Then, we will prove MCDR-V2 by reducing from VERTEX-COVER:

Assume a graph $G = (V, E)$, and calculate the value of $|E|$ in poly-time. For vertex v_i corresponding to several edges $e_1, e_2, ..e_j$, we construct a_i channels with these $d_1, d_2, ..d_j$. These channels should make sure that every d_k can be at the $1st, 2nd, ...jth$ places, so $a_i \leq j^2$, which follows poly-time.

Then, if we want to solve the VERTEX-COVER problem (is there a vertex set of size $\leq k$ which can cover all the edges). We can start at one channel, after download all the items in this channel, we can transform to another one. Because d_k can be at the $1st, 2nd, ...jth$ places, we will be able to find a proper channel, in which the item d_k is just after a switch and we don't need to wait other items before getting d_i.

Then, if we find a scenario of access-time $\leq (|E| + k - 1)$, it means we find a vertex cover set of size $\leq k$. Specially, in this case the constraint on switch-time h is infinite.

Therefore we have proved: VERTEX-COVER $\leq_P$ MCDR-V1, MCDR-V2. Then MCDR-V1 and MCDR-V2 are NP-complete.

$\square$

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