CS240 Homework 4

Shen Linfeng Student ID: 2018E8018461028

May 27, 2020

Problem 1

1) Given an assignment in which at most k variables are true, we can calculate the formula in polynomial time, so this problem is in NP.

2) Consider the SAT : Given CNF formula Φ, does it have a satisfying truth assignment?

   Let k be the number of input variables of Φ, then (can Φ be satisfied by an assignment in which at most k variables are true) is an instance of the origin problem.

3)
   - If we have yes to this SAT problem, which mean we can find an assignment with k variables to make Φ true. And in this case, the number of variables is true will be $\leqslant k$, so we can get yes to the origin problem.
   - If we have yes to the origin problem, which mean we can find an assign to make Φ true, so we can get yes to this SAT problem.

Thus this problem is at least as hard as SAT problem. Since the SAT problem is NP-complete, we have this problem is also NP-complete.

Problem 2

1) Given a simple cycle in G, we can determine whether the sum of its edge weights is zero in polynomial time. Thus Zero-Weight-Cycle is in NP.

2) Consider the Directed-Hamiltonian-Cycle problem $G = (V, E)$. Let $n = |V|$. Construct a weighted directed graph G' with $2n$ nodes, such that every $v_i \in V$ corresponds to two nodes, $u_i, w_i \in V'$.

   Add an edge from u_i to w_i with weight 1 for every $i < n$, and add an edge from u_n to w_n with weight $(1 - n)$.

   And then add edges (w_i, u_j) with weight 0 for every edge $(v_i, v_j) \in E$.

   So, (is there a simple cycle in G' so that the sum of the edge weights on this cycle is exactly 0) is an instance of Zero-Weight-Cycle Problem.

3) When there is no edge from w_i to w_j, or u_i to u_j in G, a cycle of G must be $u_a, w_a, u_b, w_b, \cdots, w_e, u_a$. With weight of every (w_i, u_j) is 0, and weight of (u_i, w_i) is positive for $i < n$. So a simple cycle of the sum-weight is 0 iff the cycle contain all edge (u_i, w_i) in G', which mean it contain all nodes of G''s.

   - So if we have yes to this Zero-Weight-Cycle problem, we can find a zero-weight-cycle in G', then this cycle must a simple cycle contain all nodes in G', and we can replace u_i and w_i with v_i to get the solution of Directed-Hamiltonian-Cycle, which mean we get yes to this Directed-Hamiltonian-Cycle problem.
   - On the other hand, if we have yes to this Directed-Hamiltonian-Cycle problem, we can get a simple cycle that contain all nodes in G, then we can replace v_i with u_i and w_i to get a zero-weight-cycle, which mean we get yes to this zero-Weight-Cycle problem.

Thus this problem is at least as hard as Directed-Hamiltonian-Cycle problem. Since the Directed-Hamiltonian-Cycle problem is NP-complete, we have Zero-Weight-Cycle is also NP-complete.

Problem 3

1) Given a one-to-one mapping $f : V' \to V$, we can check if every edge $(v_i, v_j) \in E'$ have $(f(v_i), f(v_j)) \in E$ in $O(|E|)$. Thus this problem is in NP.

2) Consider the Independent Set problem $G = (V, E)$ with k. Construct another graph $G' = (V', E')$, with V' is a set contain k nodes, and $E' = \varnothing$.

   So, (is G' equivalent to a subgraph of G) is an instance of origin problem.

3) - If G' is equivalent to a subgraph of G, then let $f : V' \to V$ be the map function. Because $E' = \varnothing$, we can know there is no edge between u and v for any $u, v \in V$, so there is no edge between $f(u)$ and $f(v)$. That mean the set $\{f(v)|\ v \in V'\}$ is an independent set of G, there is an independent set of size $\geqslant k$.
   - On the other hand, if there is an independent set of size $\geqslant k$, then select k of them to a set A. It will be easy to map nodes from A to V' one-to-one. Because there is no edge between any two nodes from A, so G' is a subgraph of G.

Thus this problem is at least as hard as Independent Set problem. Since the Independent Set problem is NP-complete, we have this problem is also NP-complete.

Problem 4

1) Given a mapping as schedule $f : C \to S$, we can calculate the number of conflicts occurs for every $r \in R$, and then check if it bigger than K. It will cost $O(|C| \cdot |R|)$. Thus this problem is in NP.

2) Consider the 3-Colorability problem: Given an undirected graph $G = (V, E)$ does there exist a way to color the nodes red, green, and blue so that no adjacent nodes have the same color?

   Construct set $C = V$, $S = \text{red}, \text{blue}, \text{green}$, $R = \{\{u, v\}|\ (u, v) \in E\}$. Let $K = 0$, and it will be an instance of the sehedule problem with input $\{C, S, R, K\}$.

3) In this situation,

   - If we get yes to this schedule problem, which mean we have a mapping f to make the number of conflicts no more than $K = 0$, which mean for every $\{u, v\} \in R$, $f(u)$ and $f(v)$ are different. Then we can use f to color G to make no adjacent nodes have the same color.
   - If we get yes to this 3-Colorability problem, which mean we have a mapping $f' : V \to S$ to make no adjacent nodes have the same color. Then we can use f' to assign C to S to make sure there will no $\{u, v\} \in R$ will conflict.

Thus this problem is at least as hard as 3-Colorability problem. Since the 3-Colorability problem is NP-complete, we have this problem is also NP-complete.

Problem 5

1) Given a simple TA cycle, we can calculate the sum of nodes that the cycle contain and determine whether it $\geqslant K$ in $O(n)$. Thus this problem is in NP.

2) Consider the Longest Path problem: Given a digraph $G = (V, E)$, does there exists a simple path of length at least k edges.

   Construct a digraph $G' = (V', E')$, with $V' = V \cup \{s\}$, and $E' = E \cup \{(s, v)|\ v \in V\} \cup \{(v, s)|\ v \in V\}$.

   Let every nodes in V' represent a TA, (v_i, v_j) mean v_i works as a TA for v_j, and $K = k + 2$.

   So, (is there a simple TA cycle containing at least K TAs on G') is an instance of the origin problem.

3) For every path in G with a nodes, which mean its length is $a - 1$, can add s to form a cycle in G' with $a + 1$ nodes. For every cycle in G' with a nodes, can remove s and related edges to get a path with $a - 1$ nodes and $a - 2$ length in G if it contain s, or remove one node and and related edges to get a pathwith $a - 1$ nodes and $a - 2$ length in G if it don't contain s.

   - So, if we find a simple TA cycle containing at least K TAs, we can get a simple path in G whose length is at least $K - 2 = k$. This is the solution of the Longest Path problem.
   - On the other hand, if we find a simple path A' in G whose length is at least k, we can get $A' \cup \{s\}$ could be a simple cycle whose length is at least $k + 2 = K$.

Thus this problem is at least as hard as Longest Path problem. Since the Longest Path problem is NP-complete, we have this problem is also NP-complete.

Problem 6

1) Given an input set S, we can check if the total weight is at most b and if the corresponding profit is at least k. It will cost $O(|S|)$.

2) Consider the Subset Sum problem: Given natural numbers set $W = \{w_1, \cdots, w_n\}$ and an integer s, is there a subset S that adds up to exactly s.

   Let $n = |W|$, $k = b = s$, construct n elements' set $A = a_1, a_2, \cdots, a_n$ where $a_i = w_i$ for every i, represent weighted of ith item, $C = c_1, c_2, \cdots, c_n$ where $c_i = w_i$ for every i, represent profit of ith item. Then (find a subset of the items with total weight at most b such that the corresponding profit is at least k) is an instance of the Knapsack problem.

3)
   - If we have yes to this Knapsack problem, there must have a set S that:

     $$\sum_{i \in S} a_i \leqslant b$$
     $$\sum_{i \in S} c_i \geqslant k$$

     Due to $a_i = c_i = w_i$ for every i, and $k = b = s$, we can get:

     $$\sum_{i \in S} w_i \leqslant b$$
     $$\sum_{i \in S} w_i \geqslant k$$

     which mean:

     $$\sum_{i \in S} w_i = s$$

     So, we also have yes to this Subset Sum problem.

   - On the other hand, if we have yes to this Subset Sum problem, which mean there is a set S that:

     $$\sum_{i \in S} w_i = s;$$

     So we can get:

     $$\sum_{i \in S} a_i = \sum_{i \in S} w_i = s = b \leqslant b$$
     $$\sum_{i \in S} c_i = \sum_{i \in S} w_i = s = k \geqslant k$$

     So, we also have yes to this Knapsack problem.

Thus this problem is at least as hard as Subset Sum problem. Since the Subset Sum problem is NP-complete, we have this problem is also NP-complete.