CS5820 HW9

Renhao Lu, NetID: rl839

November 8, 2021

1 NP-Complete proof sketches

1.1 UNSAT

The approach is not correct. Because UNSAT is not a NP problem. No matter which certificate we are provided, we can not prove that there is no other input that make Φ true.

1.2 Magnet Problem

The approach is correct. The complete proof is in below:

1.2.1 Magnet problem is in NP

Certificate: a set of strings (may contain replicate strings)
Check: if the set of strings used up all symbols in the symbol collection, the answer is yes. If the number of strings is a and the number of symbols is b, the checking time is $n(ab)$, which is polynomial.

1.2.2 Reduce Hamilton Cycle problem to Magnet problem

Convert input Given a directed graph $G = (V, E)$. For every node $v \in V$, create two letters v_1, v_2. For every edge $(u, v) \in E$, create a string $u_2 v_1$.

Convert output If Magnet problem has a yes answer, Hamilton cycle problem has a yes answer. Otherwise, the answer of Hamilton cycle problem is no.

Running time Assume $|V| = n$, $|E| = m$. We will create $2n$ symbols and m strings, and the length of all strings are 2. Hence, the total running time is $O(n + m)$, which is polynomial.

Correctness We will prove that the yes/no answer of magnet problem is equivalent to the answer of Hamilton problem.
If Hamilton problem has a solution, we note this cycle $C = \{(v_1, v_2), (v_2, v_3), ..., (v_n, v_{n-1})\}$ where $V = \{v_1, v_2, ..., v_n\}$. We know that for each node $v_i \in V$, exist exactly two edges $(v_i, v_j), (v_{j'}, v_i) \in C$. Then we created symbols v_{i1}, v_{i2} and strings as we previously described. Here, for each edge $(v_i, v_j) \in C$, we add the string $v_{i2} v_{j1}$ into our collection. Because each node v_i, appeared twice in two different edge at left side and right side respectively, each symbol v_{i1}, v_{i2} must appear in the string collection exactly once. Meanwhile, all symbol must have been used because Hamilton cycle must travel throught all nodes. Hence, the string collection would be a solution for Magnet problem.
If the Magnet problem has a solution, we note the alphabet as $A = \{v_{11}, v_{12}, v_{21}, v_{22}, ..., v_{n1}, v_{n2}\}$ the solution string collection $S = \{v_{i2} v_{j1}, v_{i'2} v_{j'1}, ...\}$. For each symbol, we have exactly one "amgnet", so each symbol must appear exactly once in S. So we know for each $v_{i1} and v_{i2}$, exist one string $v_{j2} v_{i1}$ and $v_{i2} v_{j'1}$ in S. Hence, we pick one random symbol v_{i_2} as the start point and reorginze S in a end-to-end pattern as following: $\{v_{i2} v_{j1}, v_{j2} v_{k1}, ..., v_{p2} v_{t1}, v_{t2} v_{i1}\}$. Then we create a cycle as for the solution for Hamilton cycle problem: $C = \{(v_i, v_j), (v_j, v_k), ..., (v_p, v_t), (v_t, v_i)\}$. This cycle doesn't have any replicates because no replicates exists

in S. Meanwhile C covers all nodes because S covers all symbols in A. Hence C is a correct solution for Hamilton problem.

1.3 Subway Riding

The apporach is not correct. From the porblem we can know that $\sum_{i=1}^{n} w_i \geq W$. Hence createing n cycles will need totally $(\sum_{i=1}^{n} w_i) - n + 1$ nodes in G. Creating new graph G will take at least $O(W - n)$, which is not polynomial time.

2 Disrupting an Enemy's Railway Network II

Railway problem is in P, and it can be reduced to the original railway disruption problem.

Step 1: convert input Given the input of the railway problem directed graph $G = (V, E)$, $h \in V$, $T \subseteq V$, and integer $k < |T|$. We create a new directed graph $G' = (V', E')$ for the input of original railway problem. We first add h to V'. Then For each node $v_i \in V(v_i \neq h)$, we add two nodes v_{i-in}, v_{i-out} into V', and we add edges (v_{i-in}, v_{i-out}) to E' with capacity $c_e = 1$. For each edge $(v_i, v_j) \in E$, we add edge (v_{i-out}, v_{j-in}) into E' with capacity $c_e = +\infty$. For all $v_i \in T$, we add $v_i - out$ to T'. And we set all $w_i = +\infty$ for $v_{i-out} \in T'$. Then we pass the directed graph G', the capacity of the edges $\{c_e\}$, set T', and $\{w_i\}$ as the input of the oringinal railway problem. Because all $w_i = +\infty$, we only need to find the minimized $\sum_{c \in F} c_e$ in the original railway problem.

Step 2: convert output We note the output of the original railway problem is a set F of edges to be destroyed. If $|F| \leq k$, the answer for the railway problem II is yes. Otherwise, the answer is no.

Step 3: Running time Assume for the input of railway problem II $G = (V, E)$, $|V| = n, |E| = m, |T| = p < n$.
Converting input: in $G' = (V', E')$, $|V'| = 2n - 1$, $|E'| = m + n - 1$, so building up the graph and setting the edge capacities will take $O(m + n)$. Initializing $|T'|$ and $\{w_i\}$ takes $O(p) < O(n)$, so toal would still be $O(m + n)$
Converting output: Getting the output F size can take at most $O(m + n)$, and comparing size with k is $O(1)$. So total is $O(m + n)$
Find the solution for the original railway problem I: The maximum possible flow in equal to $|T'| = |T| = p$. So the running time for the original railway problem is $O(|E'|p) = O((m + n)p) < O((m + n)n)$.
The total running time of this reductions is $O(m + n) + O(m + n) + O((m + n)n) = O((m + n)n)$, which is polynomial.

Step 3: Correctness We claim that $|F| \leq k$ from the original railway problem is equivalent to the yes answer in the railway problem II.
1. If $|F| \leq k$ from the original railway problem I, we can prove that railway problem II have a solution. We can first claim that All edges in F is (v_{i-in}, v_{i-out}) and doesn't contain any edge like (v_{i-out}, v_{j-in}), because all edges (v_{i-out}, v_{j-in}) has a capacity of $+\infty$, and they cannot be seperated by minimum s-t cut. Then we create a set of node Q, and add each v_i if $(v_{i-in}, v_{i-out}) \in F$. We can tell that $|Q| = |F| \leq k$. Because $w_i = +\infty$, all terminal nodes must be disconnected. So Q is a solution for railway problem II.
2. If railway problem II has a solution, we can prove that the solution $|F|$ not larger than k. We note $Q = \{v_{i1}, v_{i2}, ..., v_{ik}\}$ is a set of statations as the solution of railway problem II. Then we set $F_1 = \{(v_{i1-in}, v_{i1-out}), (v_{i1-in}, v_{i1-out}), ..., (v_{ik-in}, v_{ik-out})\}$. Because in railway problem II, disabling the statations in Q can disconnect all terminals in T, destroying edges in F_1 can disconnect all terminals in railway problem I as well. Because for all $(v_{i-in}, v_{i-out}), c_e = 1$, $|F|_{optimal} = min\{\sum_{c \in F} c_e\} \leq \sum_{c \in F_1} c_e = |F_1| = k$. Hence the optimal $|F| \leq k$,

3 Side gig problem

3.1 Side gig problem is in NP

Certificate: a set of jobs you plan to do during the summber $J = i_1, i_2, i_3, ...$
Check: If for any two $i, j \in J, D_i \cap D_j = \emptyset$, and $\sum_{i \in J} p_i \geq C$, the answer is yes.

3.2 Reduce independent set problem to side gig problem

Convert input Given the input of independent set problem, undirected graph $G = (V, E)$ and integer k. We assume $V = 1, 2, 3, ..., n, |V| = n$ and $E = e_1, e_2, ..., e_m, |E| = m > n$. Then we create n jobs, each job i' corresponds to a node $i \in V$, and each job has the same payment $p_i = 1$. Then we set summer break as m days. For each edge $e_p = (i, j) \in E$, we add day p into D_i and D_j. At last we set the credit card debit $C = k$. Now we can pass n, m, D_i, C as the input of side gig problem.

Convert output If the answer of side gig problem is yes, the output of independent set is also yes. Otherwise, the output of independent set is no.

Running time Converting input: Creating new job set, takes $O(n)$. Creating m days of summer break takes $O(m)$. Setting up D_i need to update 2 times for each edge, so takes $O(2m) = O(m)$. Hence, converting input totally take $O(n + m)$, which is polynomial.

Correctness We claim that the answer of independent set problem is equivalent to the answer of side gig problem.
1. If independent set problem has a solution, we note this solution as set $S = i_1, i_2, ..., i_k$, then we create a job set $P = i'_1, i'_2, ..., i'_k$. The total payment $\sum_{i \in P} p_i = k = C$, hence the credit card debit can be paied off. Meanwhile, because in the independent set S, no two node share one edge, we know that in P, no two jobs share a same day. Hence, we know set P is a solution for the side gig problem.
2. If the side gig problem has a solution, we note this soltion as a set of jobs $P = i'_1, i'_2, i'_3, ..., i'_k, ...$ Becasue the debit $C = k$, $|P| \geq C = k$. We select the first k members in P and transfor the jobs into another set of node which the jobs are corresponding to, $S = i_1, i_2, ..., i_k, |S| = k$. We claim that S is an independent set because if some two nodes $i, j \in S$ and $(i, j) = e_p \in E$, $D_i \cap D_j = p \neq \emptyset$, which is a contradiction. Hence, P is a solution for independent set problem.