

# Homework 11

## 1 Graded Problems

1. Prove that the following problem is in NPC: Given an undirected graph  $G = (V, E)$ , determine whether there is a spanning tree whose degree is not greater than  $k$ . That is, whether there is a subgraph  $G'(E', V)$ ,  $E' \subset E$ ,  $|E'| = |V| - 1$ ,  $G'$  is a connected graph and all its node degrees are less than or equal to  $k$ . (20pts)

First we need to prove finding the  $k$ -spanning-tree is in NP. Obviously, as long as a subgraph  $G'$  is given, it can be verified in polynomial time whether the degrees of its vertices are all less equal than  $k$  and whether it is a legal spanning tree. Hence the problem of 'Spanning Tree with Bounded Degree'(STBD)  $\in$  NP.

Then we need to prove finding the  $k$ -spanning-tree is NP-Hard. We use a Hamiltonian Path to Prove this, which is a NPC problem. A H-Path will visit each vertex exactly once, which will have a max degree of 2 for each vertex in the path. And as the path is opened,  $V = E + 1$ .

To prove that  $\text{H-Path} \leq_p \text{STBD}$ , We configure the STBD problem as follows: For an input of H-Path Graph  $G$ , we input the exact graph into STBD problem and set  $K=2$ . If there is a solution to the STBD problem, that means there is a spanning tree that goes through all vertices with degree less or equal to  $K=2$ . Since there are no branches as no point has 3 connections, this tree is a path that goes through all vertices which is a H-Path. Vice-Versa, if there is a H-Path in the graph, the path is also a tree with degree less than or equal to 2, and that satisfies the STBD problem.

Rubric (20pts):

- 5 pts for Proving  $k$ -spanning-tree is in NP.
- 15 pts for Proving  $k$ -spanning-tree is in NP-Hard.

2. You are given a directed graph  $G=(V,E)$  with weights on its edges  $e \in E$ . The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in  $G$  so that the sum of the edge weights on this cycle is exactly 0. Prove that this problem is NP-complete. (20pts)

Zero-weight-cycle is in NP because we can exhibit a cycle in  $G$ , and it can be checked that the sum of the edge weights on this cycle are equal to 0.

We now show that  $\text{subset sum} \leq \text{Zero-weight-cycle}$ . We are given the number  $w_1, \dots, w_n$ , and we want to know if there is a subset that adds up to exactly  $W$ . We construct an instance of the Zero-weight-cycle in which the graph has nodes  $0, 1, 2, \dots, n$ , and an edge  $(i, j)$  for all pairs  $i < j$ . The weight of the edge  $(i, j)$  is equal to  $w_j$ . Finally, there is an edge  $(n, 0)$  of weight  $-W$ .

We claim that there is a subset that adds up to exactly  $W$  if and only if  $G$  has a zero-weight-cycle. If there is such a subset  $S$ , then we define a cycle that starts at 0, goes through the nodes whose indices are in  $S$ , and then returns to 0 on the edge  $(n, 0)$ . The weight of  $-W$  on the edge  $(n, 0)$  precisely cancels the sum of the other edge weights. Conversely, all cycles in  $G$  must use the edge  $(n, 0)$ , and so if there is a zero-weight-cycle, then the other edges must exactly cancel  $-W$ , in other words, their indices must form a set that adds up to exactly  $W$ .

Rubric (20pt):

- 5 pts for Proving Zero-weight-cycle is in NP.
- 15 pts for Proving Zero-weight-cycle is in NP-Hard.

3. In a certain town, there are many clubs, and every adult belongs to at least one club. The town's people would like to simplify their social life by disbanding as many clubs as possible, but they want to make sure that afterwards everyone will still belong to at least one club.

Formally the Redundant Clubs problem has the following input and output.

INPUT: List of people; list of clubs; list of members of each club; number  $K$ .

OUTPUT: Yes if there exists a set of  $K$  clubs such that, after disbanding all clubs in this set, each person still belongs to at least one club. No otherwise.

Prove that the Redundant Clubs problem is *NP*-Complete. (20pts)

**Solution:**

- (a) We must show that Redundant Clubs is in *NP*, but this is easy: if we are given a set of  $K$  clubs, it is straightforward to check in polynomial time whether each person is a member of another club outside this set.
- (b) We prove Redundant Clubs is in *NP*-Hard by reducing from a known *NP*-complete problem, Set Cover, e.g.,  $\text{Set Cover} \leq_p \text{Redundant Clubs}$ . We translate inputs of Set Cover to inputs of Redundant Clubs, so we need to specify how each Redundant Clubs input element is formed from the Set Cover instance.

We use the Set Cover's elements as our translated list of people, and make a list of clubs, one for each member of the Set Cover family. The members of each club are just the elements of the corresponding family. To finish specifying the Redundant Clubs input, we need to say what  $K$  is:

we let  $K = F - K_{SC}$  where  $F$  is the number of families in the Set Cover instance and  $K_{SC}$  is the value  $K$  from the set cover instance. This translation can clearly be done in polynomial time (it just involves copying some lists and a single subtraction).

Finally, we need to show that the translation preserves truth values. If we have a yes-instance of Set Cover, that is, an instance with a cover consisting of  $K_{SC}$  subsets, the other  $K$  subsets form a solution to the translated Redundant Clubs problem, because each person belongs to a club in the cover. Conversely, if we have  $K$  redundant clubs, the remaining  $K_{SC}$  clubs form a cover. So the answer to the Set Cover instance is yes if and only if the answer to the translated Redundant Clubs instance is yes. This completes the reduction, and we confirm that the given problem is *NP*-Hard.

Thus this problem is NP-Complete.

Rubric (20pt):

- 5 pts for Proving Redundant Clubs is in NP.
  - 15 pts for Proving Redundant Clubs is in NP-Hard.
    - 5 pts for the claim that  $\text{Set Cover} \leq_p \text{Redundant Clubs}$ .
    - 5 pts for the construction.
    - 4 pts for the reduction proof.
    - 1 pt for the conclusion.
4. Given a graph  $G = (V, E)$  with an even number of vertices as the input, the HALF-IS problem is to decide if  $G$  has an independent set of size  $|V|/2$ . Prove that HALF-IS is in *NP*-Complete. (20pts)

**Solution:**

- (a) Given a graph  $G(V, E)$  and a certifier  $S \subset V$ ,  $|S| = |V|/2$ , we can verify if no two nodes are adjacent in polynomial time ( $O(|S|^2) = O(|V|^2)$ ). Therefore, HALF-IS  $\in NP$ .
- (b) We prove HALF-IS is in *NP*-Hard by using a reduction of the *NP*-complete problem Independent set problem (IS) to HALF-IS, e.g.,  $\text{IS} \leq_p \text{HALF-IS}$ . Consider an instance of IS, which asks for an independent set  $A \subset V$ ,  $|A| = k$ , for a graph  $G(V, E)$ , such that vertices in  $A$  disconnected from each other:
  - i. If  $k = |V|/2$ , IS reduces to HALF-IS.
  - ii. If  $k < |V|/2$ , then add  $m$  new disconnected nodes such that  $k + m = (|V| + m)/2$ , i.e.,  $m = |V| - 2k$ . Note that the modified set of nodes  $V' (= V \cup \{m \text{ new nodes}\})$  has an even number of nodes. Since the additional nodes are all disconnected from each other, they form a subset of the independent set. Therefore, the new graph  $G'(V', E')$  where  $E' = E$  has an independent-set of size  $|V'|/2$  if and only if  $G(V, E)$  has an independent set of size  $k$ .
  - iii. If  $k > |V|/2$ , then again add  $m = |V| - 2k$  new nodes to form the modified set of nodes  $V'$ . Connect these new nodes to all the other  $|V| + m - 1$  nodes. Since these  $m$  new nodes are connected to every other, none of them should belong to an independent set. Therefore, the new graph  $G'(V', E)$  has an independent-set of size  $|V'|/2$  if and only if  $G(V, E)$  has an independent set of size  $k$ .

Hence, any instance of IS  $(G(V, E), k)$ , can be reduced to an instance of HALF-IS  $(G'(V', E'))$ . This completes the reduction, and we confirm that the given problem is *NP*-Hard.

Thus this problem is NP-Complete.

Rubric (20pt):

- 5 pts for Proving Redundant Clubs is in NP.

- 15 pts for Proving Redundant Clubs is in NP-Hard.
    - 5 pts for the claim that  $IS \leq_p \text{HALF-IS}$ .
    - 3 pts for each the construction + proof (9 pts in total)
    - 1 pt for the conclusion.
5. There are  $n$  courses at USC, each of them requires multiple disjoint time intervals. For example, a course may require the time from 9am to 11am and 2pm to 3pm and 4pm to 5pm (you can assume the number of intervals of a course is at least 1, at most  $n$ ). You want to know, given a number  $K$ , if it's possible to take at least  $K$  courses. You cannot choose any two overlapping courses. Prove that the problem is NP-complete, which means that choosing courses is indeed a difficult thing in our life. Use a reduction from the Independent Set problem. (20pts)

**Solution:**

- (a) (Showing Problem in NP) The solution of the problem can be verified in polynomial time (just check the number of the courses in the solution is larger or equal to  $K$ , and they don't have time overlap), thus it is in NP.
- (b) (Showing Problem in NP-Hard) Given an independent set problem, suppose the graph has  $n$  nodes and asks if it has an independent set of size at least  $K$ . We map the instance of IS to an instance of course choosing problem by establishing an injection  $f : V \times V \rightarrow N$ , s.t.

$$f(v_i, v_j) = \begin{cases} i * n + j & \text{if } i \leq j \\ f(v_j, v_i) & \text{otherwise} \end{cases}$$

Now we can construct an instance of the course choosing problem: each course corresponds to a vertex of the graph, and if there exists an edge  $(v_i, v_j)$  in the original graph, we let the  $i$ -th courses require the  $f(v_i, v_j)$ -th hour. The problem is to determine whether we can choose  $K$  courses. Notice that, if there exists an edge  $(v_i, v_j)$  in the original graph, then the  $i$ -th course and the  $j$ -th course will jointly require the  $f(v_i, v_j)$ -th hour, which means that we can't choose these two courses at the same time. You can verify that any other courses (except  $i$ -th and  $j$ -th course) will not require  $f(v_i, v_j)$ -th hour.

If the Independent Set problem is a "yes" instance (has an independent set of size at least  $K$ ), then we can choose the corresponding courses, and they don't overlap. For the other direction, if we can choose the corresponding courses, then it follows that the independent set problem is a "yes" instance.

Thus we can reduce the independent set problem to the course choosing problem in polynomial time. Since the independent set problem is NP-Complete, the course choosing problem is in NP-Hard.

Thus the course choosing problem is NP-Complete.

Rubric (20pt):

- 5 pts for Proving Redundant Clubs is in NP.
- 15 pts for Proving Redundant Clubs is in NP-Hard.
  - 8 pts for the construction + explanation (can be any other constructions).
  - 6 pts for the proof
  - 1 pt for the conclusion.

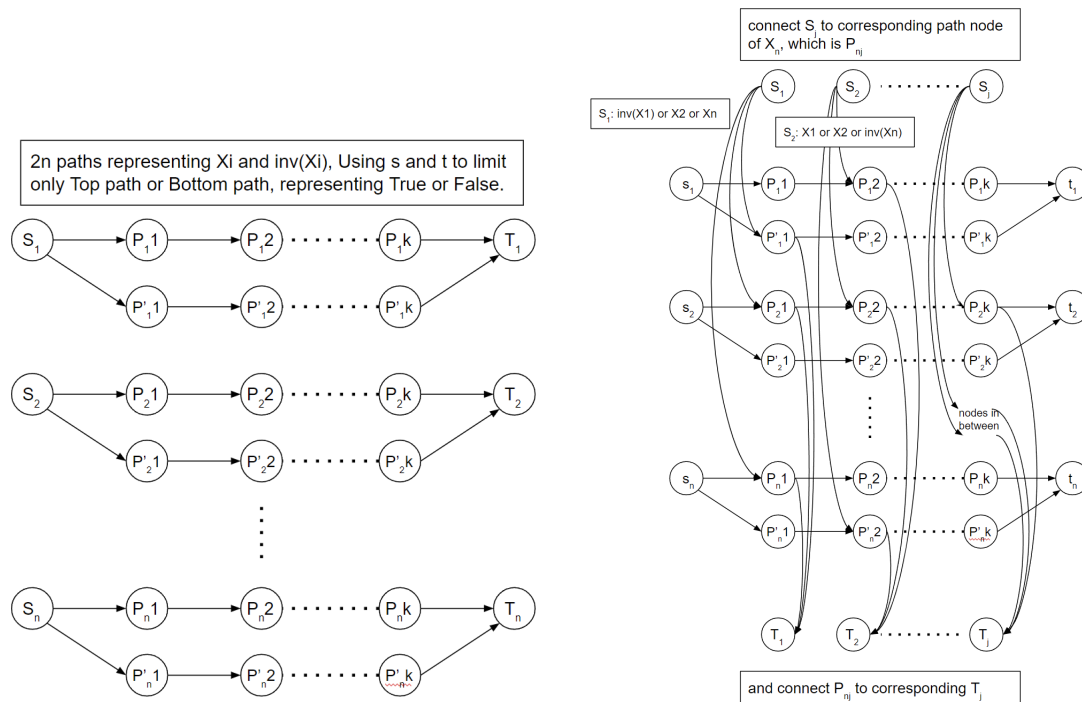
## 2 Ungraded Problems

1. The Directed Disjoint Paths Problem is defined as follows. We are given a directed graph  $G$  and  $k$  pairs of nodes  $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$ . The problem is to decide whether there exist node-disjoint paths  $P_1, P_2, \dots, P_k$  so that  $P_i$  goes from  $s_i$  to  $t_i$ . Show that Directed Disjoint Paths is NP-complete.

The problem is in NP, since we can exhibit a set of disjoint paths  $P_i$ , and it can be checked in polynomial time that they are paths in  $G$ , connect the corresponding nodes, and are disjoint.

Now we show  $3\text{-SAT} \leq_p \text{Directed Disjoint Paths}$ . Consider a 3-SAT problem given by a set of clauses  $C_1, \dots, C_k$ , each of length 3, over a set of variables  $X = x_1, \dots, x_n$ . To create the corresponding instance of the Directed Disjoint Paths problem, we will have  $2n$  directed paths, each of length  $k$ , one path  $P_i$  corresponding to variable  $x_i$  and one path  $P_i'$  corresponding to  $\bar{x}_i$ . We add  $n$  source-sink pairs corresponding to the  $n$  variables, and connect source  $s_i$  to the first node on paths  $P_i$  and  $P_i'$  and connect the last nodes in paths  $P_i$  and  $P_i'$  to sink  $t_i$ . Note that there are two directed paths connecting  $s_i$  to  $t_i$ : the path  $s_i, P_i, t_i$ , and the path  $s_i, P_i', t_i$ . We will think of selecting the first of these paths as setting the variable  $x_i$

to false (as the variable through the copies of  $\bar{x}_i$  are left unused), and selecting the second path will correspond to setting the variable  $x_i$  to true.



Now we will add  $k$  additional source-sink pairs, one corresponding to each clause  $C_j$ . Let  $S_j$  and  $T_j$  be the source-sink pair corresponding to clause  $C_j$ . We will claim that there is a path from  $S_j$  to  $T_j$  disjoint from the path selected to connect the  $s_i$ - $t_i$  source-sink pairs if and only if clause  $C_j$  is satisfied by the corresponding assignment. Assume clause  $C_j$  contains the literals  $t_{j1}, t_{j2}$  and  $t_{j3}$ . Now we have a path  $P_i$  or  $P'_i$  corresponding to each of these variables or negated variables. The paths have  $n$  nodes each, let  $v_{j1}, v_{j2}$  and  $v_{j3}$  denote the  $j$ th node on the 3 corresponding paths. We add the edges  $(S_j, v_{jl})$  and  $(v_{jl}, T_j)$  for each of  $l = 1, 2, 3$ .

Now we claim that the resulting directed graph has node-disjoint paths connecting the source-sink pairs  $s_i - t_i$  and  $S_j - T_j$  for  $i = 1, 2, \dots, n$  and  $j = 1, \dots, k$  if and only if the 3-SAT instance is satisfiable. One direction is easy to see: if the 3-SAT instance is satisfiable, then select the paths connecting  $s_i$  to  $t_i$  corresponding to the satisfying assignment, as suggested above. Then the source-sink pair  $S_j$  and  $T_j$  can be connected through the path using the true variable in the clause.

Finally, we need to show that if the disjoint paths exist, then the 3-SAT formula has a satisfying assignment. Note that the paths  $P_i$  and  $P'_i$  are disjoint, and the graph has no edges connecting different paths. The only edges outside these paths in the graph are edges entering one of the sinks, or leaving a source. As a result, the only paths in the graph connecting an  $s_i - t_i$  pair are the two paths  $s_i, P_i, t_i$  and  $s_i, P'_i, t_i$ , and the only paths in  $G$  connecting  $S_j - T_j$  pairs are the three possible paths through each of the 3 variable nodes in  $C$ . Hence, sets of disjoint paths connecting the source-sink pairs, correspond to satisfying assignments.