CSCI 570 Summer 2023 HW 7 Solutions

Graded Problems

Problem 1

Consider the vertex cover problem that restricts the input graphs to be those where all vertices have even degree. Call this problem VC-EDG. Show that VC-EDG is NP-complete. (15pts)

Hint: To show NP-hardness, reduce from VC, i.e., think of a construction where given the input of VC, i.e., a graph G and a number K, you can construct G', K', (where G' has even degrees for all its vertices), such that G has a vertex cover of size K if and only if G' has a vertex cover of size K'.

Solution:

First we need to prove even degree vertex cover problem is in NP.

The certifier takes a subset of vertices as its certificate. It verifies that the subset is of size at most K and each of the original graph's edges has one of its endpoints in the given subset. Thus problem is in NP (the same certifier to the original vertex cover problem).

Then we need to prove VC-EDG is NP-Hard.

We claim that it is polynomial time reducible from the original vertex cover problem. Let $(G = (V, E), k)$ to be an input instance of Vertex Cover. Because each edge in E contributes a count of 1 to the degree of each of the vertices with which it connects, the sum of the degrees of the vertices is exactly $2|E|$, an even number. Hence, there is an even number of vertices in G that have odd degrees.

Let U be the subset of vertices with odd degrees in G.

Construct a new instance $(\bar{G} = (V_0, E_0), k + 2)$ of Vertex Cover, where $V_0 = V \cup \{x, y, z\}$ and $E_0 = E \cup \{(x, y), (y, z), (z, x)\} \cup \{(x, v)|v \in U\}$. That is, we make a triangle with three new vertices, and then connect one of them (say x) to all the vertices in U. The degree of every vertex in V_0 is now even. Since a vertex cover for a triangle is of (minimum) size 2, it can be shown that $\bar{G}$ has a vertex cover of size $k+2$ if and only if G has a vertex cover of size k.

Hence, vertex cover with only even degree vertices is NP Complete.

Rubric (15 pts)

- 6 pt: state the problem is in NP and mention how to verify it in polynomial time by checking if it is a valid vertex cover with even degree.
- 3 pt: Choose an NP-complete problem and say that we need to prove that chosen problem $\leq_p$ even degree vertex cover. Other reduction can also get 3 points. Note: wrong direction will get 0.
- 6 pt: Showing the polynomial reduction with details and its proof.

Problem 2

Consider the ***partial*** satisfiability problem, denoted as 3-Sat(α) defined with a fixed parameter α where $0 \leq \alpha \leq 1$. As input, we are given a collection of k clauses, each of which contains exactly three literals (i.e. the same input as the 3-SAT problem from lecture). The goal is to determine whether there is an assignment of true/false values to the literals such that at least αk clauses will be true. Note that for $\alpha = 1$, we require all k clauses true, thus 3-Sat(1) is exactly the regular 3-SAT problem.

Prove that 3-Sat($\frac{15}{16}$)is NP-complete. (20 points)

Hint: If x, y, and z are variables, note that there are eight possible clauses containing them: $(x \vee y \vee z), (!x \vee y \vee z), (x \vee !y \vee z), (x \vee y \vee !z), (!x \vee !y \vee z), (!x \vee y \vee !z), (x \vee !y \vee !z), (!x \vee !y \vee !z)$ Think about how many of these are true for a given assignment of x, y, and z.

Solution:

To prove it's in NP: given a truth value assignment, we can count how many clauses are satisfied and compare it to $15k/16$.

To prove it's NP-hard:
We will show that 3-SAT $\leq_p$ 3-SAT(15/16). For each set of 8 original clauses, we add 3 NEW VARIABLES, and create all the 8 possible clauses on these 3 new variables. If the number of clauses is a multiple of 8, then we have created m new clauses for the m existing ones. Now, any assignment will satisfy only 7/8 of the new clauses by construction, so we can say, all of the original clauses in a valid solution can be satisfied if and only if 15/16 of all the new clauses can be satisfied (convince yourself by proving the previous statement both ways). Next, if the number of clauses is not a multiple of 8, say it is of the form $8a+b$ with $b < 8$. Then we follow the procedure above for the first a groups of 8 clauses and end, leading to a total of 16a + b clauses in the new instance. Remember that regardless

of any assignment, exactly 7a of the new clauses can be satisfied. Then, it can be shown that one can achieve the factor of 15/16 satisfiability in the new instance if and only if all 8a + b of the original ones can be satisfied (using b¡8 and a bit of algebra).

So, 3-Sat(15/16) is in NP and is NP-hard which makes it NP-Complete.

Example: If original given formula is
$(a \vee b \vee c) \wedge (!a \vee b \vee c) \wedge (a \vee !b \vee c) \wedge (a \vee b \vee !c) \wedge (!a \vee !b \vee c) \wedge (d \vee e \vee f) \wedge (g \vee !b \vee !c) \wedge (!a \vee !h \vee !c)$
So we add our 8 new clauses so that formula now contains total 16 clauses.i.e.:
$(a \vee b \vee c) \wedge (!a \vee b \vee c) \wedge (a \vee !b \vee c) \wedge (a \vee b \vee !c) \wedge (!a \vee !b \vee c) \wedge (d \vee e \vee f) \wedge (g \vee !b \vee !c) \wedge (!a \vee !h \vee !c) \wedge (x \vee y \vee z) \wedge (!x \vee y \vee z) \wedge (x \vee !y \vee z) \wedge (x \vee y \vee !z) \wedge (!x \vee !y \vee z) \wedge (!x \vee y \vee !z) \wedge (x \vee !y \vee !z) \wedge (!x \vee !y \vee !z)$
If the original 8 can be satisfied with some assignment, then additionally setting any assignment to the new variables satisfies 7 of the new clauses, thus 15/16 of them in the new instance. If the original number of clauses in a formula is not a multiple of 8, say it was 13, then we still only add the above 8 new clauses making a total of 21. 15/16 fraction of that is $19\frac{11}{16}$, so satisfying 19 clauses doesn't get us above the required ratio, and we must satisfy 20. Since exactly 7 of the newly added 8 can be satisfied, the total of 20 can be achieved if and only if all the original 13 can be satisfied.

Rubric (20 pts)

- 5 pt: state the problem is in NP and mention how to verify it in polynomial time by checking how many clauses are satisfied.
- 5 pt: Choose an NP-complete problem and say that we need to prove that chosen problem $\leq_p$ 3-SAT (15/16). Other reduction can also get 3 points. Note: wrong direction will get 0.
- 10 pt: Showing the polynomial reduction with details and its proof including:
  5 pt: instance construction 5 pt: using the construction proof the polynomial reduction.

Problem 3

(20pts) Given a graph $G = (V, E)$ and two integers k, m, a clique is a subset of vertices such that every two distinct vertices in the subset are adjacent.

a) The Clique problem asks: Given a graph G, and a number $k \geq 0$, does G have a clique of size k. Show that this problem is NP-complete. Hint: Reduce from Independent Set.

b) The *Dense Subgraph* Problem is to determine if given graph G, and numbers $k, m \geq 0$, does there exist a subgraph $G' = (V', E')$ of G, such that V' has at most k vertices and E' has at least m edges. Prove that the *Dense Subgraph* Problem is NP-Complete.

Solution: Proving these problems are in NP, is trivial, so here we focus on proving their NP-hardness.

First, we prove that the Independent set problem $\leq_p$ Clique $\leq_p$ Dense Subgraph Problem. Given a graph $G(V, E)$ and an integer k, an independent set decision problem outputs yes, if the graph contains an independent set of size k. For an arbitrary graph $G = (V, E)$ of n vertices, we first get the complementary graph G_c of G. G_c contains an edge between two vertices if there was no edge between them in G, and vice versa. Thus, an independent set in G, i.e., one where each pair of vertices did not have an edge, becomes a clique in G_c and vice versa. This proves that G_c has a clique of size k if and only if G has an independent set of size k. This shows Independent set problem $\leq_p$ Clique.

Further, note that a clique of size k will always contain $k(k-1)/2$ edges. Thus, if we set m to $k(k-1)/2$ then, we can say that a graph has a clique of size k if and only if the same graph has a dense subgraph with at most k nodes and at least m edges (prove both directions of the statement clearly for completeness). This shows that Clique $\leq_p$ Dense Subgraph Problem.

Example: Consider a graph $G = (V, E)$ where $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (2, 3)\}$. Suppose we want to determine if there's an independent set in G of size k=3. We first generate G_c, where $E_c = \{(1, 3), (1, 4), (2, 4), (3, 4)\}$. Now, $V' = \{1, 3, 4\}$ is an independent set of size 3 in G, and a clique of size 3 in G_c. Further, if we set $k = 4$ and $m = 4 \cdot 3/2 = 6$, then there's a clique of size 4 if and only if there is a dense subgraph with 4 vertices and 6 edges (the answer is no to both here).

Rubric (20 pts)

- 3 pt: state the problem is in NP and mention how to verify it in polynomial time.
- 5 pt: Choose an NP-complete problem and say that we need to prove that chosen problem $\leq_p$ Dense Subgraph problem. Other reduction can also get 3 points. Note: wrong direction will get 0.
- 12 pt: Showing the polynomial reduction with details and its proof including:
  4 pt: instance construction 8 pt: proving the claim. Each direction is 4 pts.

Problem 4

There are n courses at USC, each of them scheduled in 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 that there is a fixed set of possible intervals). You want to know, given n courses with their respective intervals, and a number K, whether it's possible to take at least K courses with no two overlapping (two courses overlap if they have at least one common time slot). Prove that the problem is NP-complete. (20 points)

Hint: Use a reduction from the Independent Set problem to show NP-hardness

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 constructing a course C_u for each vertex u, a time interval T_e for each edge e and letting a course C_u to consist of all intervals T_e such that e is incident on u. Then vertices u and v are disjoint if and only if courses C_u and C_v are non-overlapping. Thus, there is an independent set of size k if and only if we can choose k courses with no two overlapping. 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 (20 pts)

- 5 pts: Proving course-choosing problem is in NP
- 15 pts: Proving course-choosing-problem is in NP-Hard
  - 8 pts: the construction + explanation (can be any other constructions)
  - 6 pts: the proof
  - 1 pt: the conclusion

Ungraded Problems

Problem 1

True/False Questions (9pts)

State True or False for the following sentences and give a brief explanation.

a If someone proves $P = NP$, then it would imply that every decision problem can be solved in polynomial time.

b Assume P $\neq NP$. Let A and B be decision problems. If $A \in NPC$ and $A \leq_p B$, then $B \in$ P.

Problem 2

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. The Redundant Clubs problem is to find whether there exist K redundant clubs after removing which every person is in at least one of the remaining clubs. Prove that the Redundant Clubs problem is NP-Complete. (20pts)

Problem 3

There are N cities, and there are some undirected roads connecting them, so they form an undirected graph $G(V, E)$. You want to know, given K and M, if there exists a subset of cities of size K, and the total number of roads between these cities is larger or equal to M. Prove that the problem is NP-Complete.

Solution:

We can formalize the problem as, given an undirected graph $G(V, E)$ and asks for if there exist a subset S of K vertices such that the number of edges in the induced graph $G[S]$ is at least M.

a (Showing Problem in NP, 5 points) The solution of the problem can be easily verified in polynomial time, just check the total number of roads between these cities. Thus it is in NP.

b (Showing Problem in NP-Hard, 15 points) Given a decision version of the independent set problem, suppose it asks for whether an independent set exists of size K. Then we ask whether the complement of the graph has a subset S of K vertices such that number of edges in the induced graph $G[S]$ is at least $K(K-1)/2$. If so, we find an independent set of size K in the original graph. For the other direction, if there is an independent set of size K in the original graph, we can find a subset S of K vertices such that number of edges in the induced graph $G[S]$ is at least $K(K-1)/2$. Thus we reduced the independent set problem to this problem in polynomial time. Since independent set problem is NP-Complete, so this problem is in NP-Hard.

Thus this problem is NP-Complete.

Problem 4

Suppose we have a variation on the 3-SAT problem called Min-3-SAT, where the literals are never negated. Of course, in this case it it possible to satisfy all clauses by simply setting all literals to true. But, we are additionally given a number k, and are asked to determine whether we can satisfy all clauses while setting at most k literals to be true. Prove that Min-3-SAT is NP-Complete.

Solution:

a For a truth assignment, we can simply count the number of literals set to true. Then evaluate each clause with the truth assignment. If all clauses equal to true and at most k literals are set to true, then answer yes. So Min-3-SAT $\in$ NP.

b We reduce from vertex cover to Min-3-SAT. For any given instance of the vertex cover problem, we can construct an equivalent Min-3-SAT problem with variables for each vertex of a graph. Each edge (u, v) of the graph can be represented by a clause $(u \vee u \vee v)$ or $(u \vee v \vee v)$ which can be satisfied only by including either u or v among the true variables of the solution. For the constructed Min-3-SAT problem, there is a satisfying assignment within k true variables if and only if there is a vertex cover within k vertices to the corresponding vertex cover problem. Therefore, Min-3-SAT is NP-Hard.

Thus this problem is NP-Complete.

Problem 5

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., IS $\leq_p$ 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 are 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.

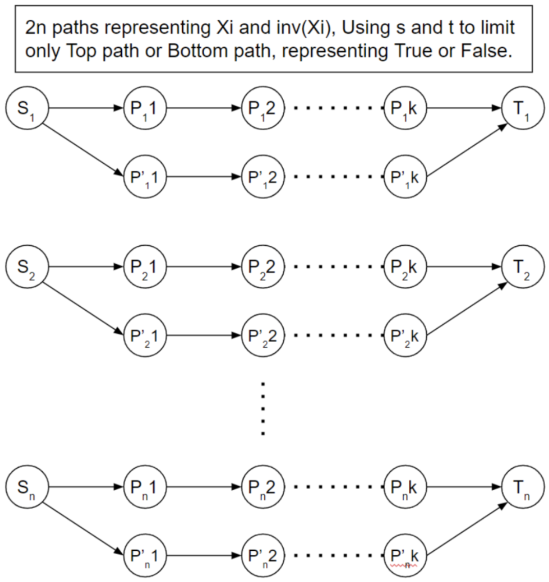
Thus this problem is NP-Complete.

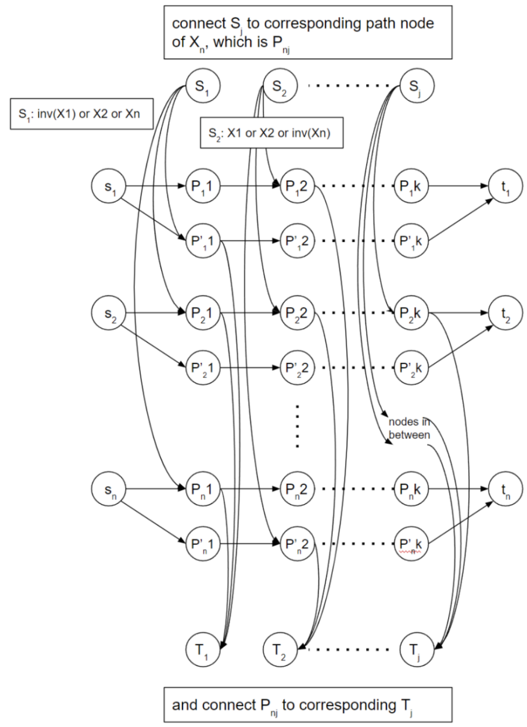
Rubric (20 pts)

- 5 pts: Proving Redundant-Clubs is in NP.
- 15 pts: Proving Redundant-Clubs is in NP-Hard
  - 5 pts: the claim that IS $\leq_p$ HALF-IS
  - 3 pts: each of the constructions + proof (9 pts in total)
  - 1 pt: the conclusion

Problem 6

The Directed Disjoint Paths Problem is defined as follows. We are given a directed graph G and k pairs of nodes $(s_1, t_1), (s_2, t_2), ..., (s_k, t_k)$. The problem is to decide whether there exist node-disjoint paths $P_1, P_2, ..., P_k$ so that P_i goes from s_i to t_i. Show that Directed Disjoint Paths is NP-complete.
Solution: 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-SAT $\leq_p$ Directed Disjoint Paths. Consider a 3-SAT problem given by a set of clauses $C_1, ..., C_k$, each of length 3, over a set of variables $X = x_1, ..., x_n$. 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 $\overline{x_i}$. 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 $\overline{x_i}$ 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 $s_j - t_j$ source-sink pairs if and only if clause C_j is satisfied by the corresponding assignment. Assume clause C_j contains the literal 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 $i = 1, 2, ..., n$ and $j = 1, ..., k$ if and only if the 3-SAT instance is satisfiable. One direction is easy to see: if the 3-SAT instance is satisfable, 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_j 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 a $s_i - t_i$ 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.

Problem 7

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

Solution:
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 H-Path$\leq$ 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. Vise-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 (20 pts)

- 5 pts: Proving k-spanning-tree is in NP
- 15 pts: Proving k-spanning-tree is in NP-Hard

Problem 8

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)

Solution: 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 subset sum $\leq$ Zero-weight-cycle. We are given the number $w_1, ..., w_n$, 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 $0, 1, 2, ..., n$, 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 (20 pts)

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