CSCI 570 - Fall 2021 - HW 10

Due November 18th

Graded

1. (20 pts) Consider the partial satisfiability problem, denoted as 3-Sat(α). We are given a collection of k clauses, each of which contains exactly three literals, and we are asked 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 3-Sat(1) is exactly the 3-SAT problem from lecture.

   Prove that 3-Sat(15/16)is NP-complete.

   Hint: If x, y, and z are literals, there are eight possible clauses containing them: (x ∨ y ∨ z), (!x ∨ y ∨ z), (x ∨ !y ∨ z), (x ∨ y ∨ !z), (!x ∨ !y ∨ z), (!x ∨ y ∨ !z), (x ∨ !y ∨ !z), (!x ∨ !y ∨ !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 ≤p 3-SAT(15/16). For each set of 8 original clauses, create 8 new clauses using 3 new variables, and construct the clauses considering the collection of all possible clauses on the 3 new variables. If the number of clauses is a multiple of 8, then we are done: any assignment will satisfy 7/8 of the new clauses, so we must satisfy all of the original clauses in a valid solution to satisfy exactly 15/16 of the clauses. If the number of clauses is not a multiple of 8, we will satisfy more than 15/16 of the clauses, but if even one of the original clauses is not satisfied, we will have satisfied less than 15/16 of the clauses.

   So, 3-Sat(15/16) is in the intersection of NP and NP-hard which is the class NP-Complete

   Example:

If original given formula is

(a ∨ b ∨ c) ∧ (!a ∨ b ∨ c) ∧ (a ∨ !b ∨ c) ∧ (a ∨ b ∨ !c) ∧ (!a ∨ !b ∨ c) ∧ (d ∨ e ∨ f) ∧ (g ∨ !b ∨ !c) ∧ (!a ∨ !h ∨ !c)

So we add our 8 new clauses so that formula now contains total 16 clauses.

i.e. : (a ∨ b ∨ c) ∧ (!a ∨ b ∨ c) ∧ (a ∨ !b ∨ c) ∧ (a ∨ b ∨ !c) ∧ (!a ∨ !b ∨ c) ∧ (d ∨ e ∨ f) ∧ (g ∨ !b ∨ !c) ∧ (!a ∨ !h ∨ !c) ∧ (x ∨ y ∨ z) ∧ (!x ∨ y ∨ z) ∧ (x ∨ !y ∨ z) ∧ (x ∨ y ∨ !z) ∧ (!x ∨ !y ∨ z) ∧ (!x ∨ y ∨ !z) ∧ (x ∨ !y ∨ !z) ∧ (!x ∨ !y ∨ !z)

And as described above there will be 15/16 satisfied clauses.

Note we add for every 8 original clauses, new 8 clauses. But if original number of clauses in a formula is not a multiple of 8, then there will be more than 15/16 clauses satisfied. Which is okay for us, since we want AT LEAST 15/16.

2. (20 pts) Consider modified SAT problem, SAT' in which given a CNF formula having m clauses in n variables $x_1, x_2, \ldots, x_n$, the output is YES if there is an assignment to the variables such that exactly m – 2 clauses are satisfied, and NO otherwise. Prove that SAT' is also NP-Complete.

   Solution:

   To show that SAT' is NP-Complete,

   First we will show that SAT' ∈ NP:

   Given the assignment values as certificate, we can evaluate the SAT' instance and verify if it is satisfied. This is same as the SAT-verification. Moreover, we can count the number of satisfied clauses and check if it is equal to m - 2 in linear time.

   Next, we show that SAT ≤p SAT':

   Construction: Add four more clauses $x_1, x_2, \neg x_1, \neg x_2$ to the original SAT instance.

   Claim: CNF formula obtained for SAT', F' has an assignment which satisfies SAT' iff CNF formula of SAT, F has an assignment which satisfies SAT.

   =>) if F has an assignment which satisfies SAT, then F' has an assignment which satisfies SAT'

   Proof: If an assignment $x_1 \ldots x_n$ satisfies F, then it satisfies exactly two of the four extra clauses, giving exactly m + 2, which is nothing but m' – 2 satisfied clauses for the F'.

   <=) if F' has an assignment which satisfies SAT', then F has an assignment which satisfies SAT

   Proof: By construction, the only unsatisfied clauses for F' must be one of x_1 or $\neg x_1$ and one of x_2 or $\neg x_2$, so all the original m clauses are satisfied.

   Example:

   If given SAT instance F is

   $(a \vee b \vee c) \wedge (d) \wedge (a \vee \neg b)$

   Then SAT' instance after construction F' will be

   $(a \vee b \vee c) \wedge (d) \wedge (a \vee !b) \wedge (x_1) \wedge (\neg x_1) \wedge (x_2) \wedge (\neg x_2)$

3. (20 pts) Given a graph G=(V,E) and two integers k, m, the *Dense Subgraph Problem* is to find a subset V' of V, whose size is at most k and are connected by at least m edges. Prove that the *Dense Subgraph Problem is* NP-Complete.

Solution:

Proving this problem NP is trivial, So here we show how to prove it's NP-Completeness.

We prove that the Independent set problem $\leq_P$ 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

A clique is a subset of vertices of an undirected graph G such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete. We know that a clique will always contain k*(k-1)/2 edges if there're k vertices in G, and that an independent set in G is a clique in G_c (the complement graph of G) and vice versa.

Then we set m to k*(k-1)/2 and test with the dense subgraph problem.

Claim: There exists an independent set of size k in G (equivalently, a clique in G_c of size k), iff there exists a subgraph of G_c with at most k vertices and at least m = k*(k-1)/2 edges.

=>) If there exists a clique in G_c of size at least k, then there exists a subgraph of G_c with at most k vertices and at least k*(k-1)/2 edges.

If there is a clique of size at least k then there is a clique of size exactly k. Moreover, by definition, a clique of size k would have k * (k-1)/2 edges.

<=) If there exists a subgraph of G_c with at most k vertices and at least k*(k-1)/2 edges, then there is a clique of size at least k.

For a subgraph to have k * (k-1) edges, implies there are k vertices. So this subset with k vertices forms a clique in G_c of size k.

Example: Consider a graph G=(V,E) where V={1,2,3,4} and E={(1,2), (2,3)}. To determine if there's a independent set in G of size 3, we first generate G_c, where E_c = {(1,3),(1,4),(2,4),(3,4)}. Since there's a dense graph with 3 vertices and 3*2/2=3 edges in G_c, which is V'={1,3,4}, E'={(1,3),(1,4),(3,4)}, we can say there's an independent set in G of size 3. On the other hand, if we set k=4 and m=4*3/2=6, then there's no such dense graph in G_c, thus no independent set of size 4 in G.

Ungraded

4. (20 pts) (Modified from Textbook 8.16)Consider the problem of reasoning about the identity of a set from the size of its intersections with other sets. You are given a finite set U of size n, and a collection $A_1 \ldots A_m$ of subsets of U. You are also given numbers $c_1 \ldots\ldots c_m$, and numbers $d_1 \ldots\ldots d_m$. The question is:

   Does there exist a set $X \subset U$ so that for each i = 1…m, the cardinality of $X \cap A$ is larger than c_i but smaller than d_i? We will call this an instance of the *Intersection Inference Problem, with* input U, $\{A_i\}$, and $\{c_i\}$, $\{d_i\}$. Prove that Intersection Inference is NP-complete.

   Solution:

   To show this problem is NP-Complete, we first prove it's NP.

   Given the set $X \subset U$, we can enumerate all Bi in the collection, calculate the intersection with X, and compare its cardinality with c_i and d_i. So this problem is NP.

   Then, we prove that the Dominating Set ≤P Intersection Inference. For an arbitrary graph G = (V, E) of n vertices, we let V to be set U, and construct n+1 subsets $A_1 \ldots A_{n+1}$ by: A_i (1<=i<=n) consists of all adjacent nodes of $v_i \in V$, and $A_{n+1} = U$. Then We set $c_1 \ldots c_n$ to 1, $d_1 \ldots d_n$ to infinity, c_{n+1} to 0, d_{n+1} to k. The solution of the intersection inference problem with this inputs is exactly the dominating set of G of size no more than k.

   (If you are not aware of Dominating Set Problem, here's a description: https://en.wikipedia.org/wiki/Dominating_set . It's very easy to get confused with Vertex Cover Problem, like what I did :)

   Example: Lets assume we have a graph G=(V,E) where V={1,2,3,4} and E={(1,2), (2,3)}. To find if there's a dominating set of size no more than 2, we build up 5 sets:

   A1={1,2}, A2={1,2,3}, A3={2,3}, A4={4}; A5={1,2,3,4}

   and set c1=c2=c3=c4=1, c5=2; d1=d2=d3=d4=0, d5=2

   the result of the Intersection Inference would be X={2, 4}, which corresponds to a dominating set of G of size 2.

5. (20 pts) (Textbook 8.28) The following is a version of the independent Set Problem. you are given a graph G = (V, E) and an integer k. For this problem, we will call a set $I \subset V$ *strongly independent if,* for any two nodes v, u ∈ I, the edge (v, u) does not belong to E, and there is also no path of two edges from u to v, that is, there is no node w such that both (u, w) ∈ E and (w, v) ∈ E. The Strongly independent Set Problem is to decide whether G has a strongly independent set of size at least k.

   Prove that the Strongly independent Set Problem is NP-complete.

   Solution:

To show this problem is NP-Complete, we first prove it's NP.

Given a vertex subset V' $\subseteq V$, we can check in polynomial time if any pair of vertices u, v $\in V'$ are either connected by an edge in E for by another node $w \in$ V. Thus we can determine if a vertex subset if a strong independent set in polynomial time.

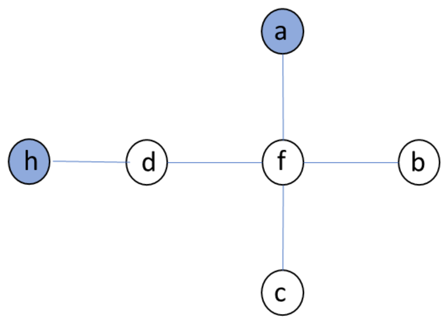
Then, We prove that Independent Set $\leq_P$ Strong Independent Set.

For an arbitrary graph G = (V,E), we split every edge in E into two nodes and connect them by a new node w_e (that means, an edge (u,v) in E becomes two edges (u,w_e) and (v,w_e). Then we connect all w_e nodes pairwise. Mathematically, we generate a new graph G+=(V+, E+) where V+ = V $\cup$ {w_e | e $\in$ E}, and E+ = {(u,w_e), (v,w_e) | e=(u, v), e $\in$ E} $\cup$ {(w_e ,$w_{e'}$) | e, e' $\in$ E}. To determine if there's an independent set in G of size at least k, we can check if there's any strong independent set in G+ with the same condition. Then:

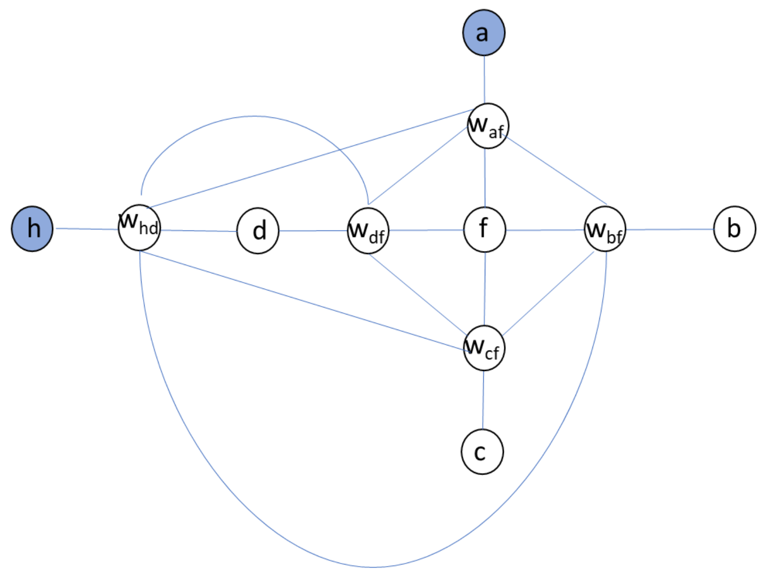
1.If there is not such set: It's easy to prove that any independent set of G is also a strong independent set of G+, so there's also no independent set of size at least k in G.

2. If there is such set: Let V+' be that set. Since V+' is a strong independent set of G+, if any node in x $\in$ {w_e | e $\in$ E} is in V+', then V+' can only contain x (because all other nodes in V+ are all not further than 2 to x). In this case, |V+'| = 1, and it's trivial to find an independent set of size 1 in G. On the other hand, if V+' don't intersect with {w_e | e $\in$ E}, then V+' $\subseteq$ V, making it an independent set of V. In both cases, there's an independent set in G of size at least k.

Example:



G



G+