

Homework 5

February 2020

1 Efficient Yet Representative Meetings

Given an undirected graph $G = (V, E)$. Suppose we aim to know whether there is a subset of the vertices $D \subseteq V$ with $|D|$ at most k , k integer, such that for all $v \in V \setminus D$ there exists some $u \in D$ such that $(u, v) \in E$.

This problem could arise in a large organisation that wants to make their meetings more efficient by having fewer people attend, whilst having input and updates from all ongoing projects. The members can be seen as the nodes and there are edges (u, v) between two members u and v if they collaborate on the same project. The aim is to have at least one representative of each collaboration attend these meetings whilst keeping the number of attendees to be less or equal to k . We will call this problem EYR, Efficient Yet Representative meeting problem.

1. Show that the defined problem is in NP, i.e.: solutions to EYR can be verified in polynomial time. (The background to this part will be introduced in the Wednesday lecture.)
2. Give a polynomial-time reduction from the Set Cover problem to EYR. What does this entail about its complexity?

To provide the reduction, break it down into the following steps:

- (a) Provide a translation from Set Cover to EYR: create an instance of EYR given the input to Set Cover (a collection of subsets S_1, \dots, S_m of a set U and an integer k). Show this translation is polynomial in time with respect to Set Cover's input size.
 - (b) Show that if the translated instance of EYR has an adequate $D \subseteq V$, then U can be covered by at most k of the subsets in Set Cover.
 - (c) Show that if U can be covered by at most k of the subsets in Set Cover, then the translated instance of EYR has an adequate $D \subseteq V$.
3. (*Optional*) Can you relate this problem to any of the other decision problems we have seen so far? We do not ask for another reduction, just loose ideas, i.e., you can test your intuition.

A couple of things to keep in mind when solving problems involving reductions:

1. When translating a problem A into an instance of another problem B, it might not be immediately evident that the translation will work. If you feel that what you've come up with looks a bit unusual, a good way to proceed is to try to prove that there is a solution to problem A if and only if the translated instance of problem B has one (i.e.: in the context of NP-complete decision problems, problem A gives an answer "yes" if and only if the translated instance of problem B gives an answer "yes").

This will help clear out doubts: if we manage to complete the proof, the translation gives rise to a correct problem reduction; if it doesn't, we might discover what is missing or should be changed in the translation.

2. Remember that in giving the reduction from problem A to B, the resulting translated input to B often belongs to a subspace of the whole input space to B. That is, we have inputs with a more specific shape or form than problem B would in general take in. This input might have some property that we can then make the most of in finishing the reduction: namely, this property might be vital in proving the equivalence of an instance of A and its translated instance of B.

For example, in this week's homework, when reducing set cover to the proposed problem EYR, the graph resulting from your translation will not be any arbitrary graph. Keep in mind: is there a property arising from your construction that can help justify the instance equivalence?