Algorithms and Complexity. Exercise session 6

NP-problems

Frequency Allocation In mobile telephony, you need to solve the *frequency allocation* problem, which is stated as follows. There are a number of transmitters deployed and each of them can transmit on any of a given set of frequencies. Different transmitters have different frequency sets. Some transmitters are so close that they can not transmit at the same frequency, because then they would interfere with each other. (This is actually not dependent on geographical distance - if can be a mountain, a house or other structure.)
You are given the frequency range of each transmitter and the pairs of transmitters can interfere if they send in the same frequency. The problem is to determine if there is any possible choice of frequencies so that no transmitter interferes with any other. Formulate this problem as a graph problem and prove that it is NP-complete!

Solution to Frequency Allocation

a) Formulate the frequency allocation problem as a graph problem. Let the vertices correspond to transmitters and edges correspond to interference between transmitters. Every vertex is labeled with a frequency range F_i. The question is whether one can allocate to each vertex a frequency from its frequency range so that no vertices are connected with an edge having the same frequency.

b) Show that the frequency allocation problem is in NP. Guess (non-deterministic) a frequency assignment. Go through each vertex and verify that its frequency is in the frequency set. Go also through each edge and verify that the endpoint of the frequencies are different. This takes linear time in the size of the graph.

c) Show that the frequency allocation problem is NP-hard.

Reduce k-coloring problem to frequency allocation:

k-coloring(G, k) =
for each vertex v_i in the graph G
$F_i \leftarrow \{1, \ldots, k\}$
return FrequencyAllocation$(G, \{F_i\})$

Now, show that there is a k-coloring of graph G iff there is a correct assignment of frequencies to G, where every vertex has frequency set $\{1, \ldots, k\}$.

Suppose we have a k-coloring of G. Number the colors from 1 to k. If a vertex has color i, we assign to the corresponding vertex (transmitter) in the frequency allocation problem the frequency i. This is a correct frequency assignment because we have been based on a correct k-coloring.

In the other direction: assume that we have a correct frequency allocation. We get a k-coloring by allowing a vertex to have color i if the corresponding transmitter have been assigned frequency i.

□

Hamiltonian path in a graph Show that the HAMILTONIAN PATH problem is NP-complete. The problem is to determine if there is a simple path that crosses each vertex of the graph.

Solution to Hamiltonian path in a graph

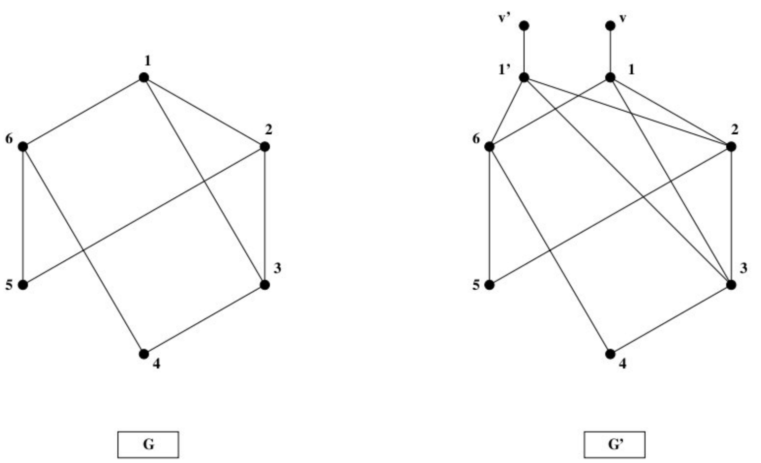
A Hamiltonian path is a simple open path that contains each vertex in a graph exactly once. The Hamiltonian Path problem is the problem to determine whether a given graph contains a Hamiltonian path.

To show that this problem is NP-complete we first need to show that it actually belongs to the class NP and then find a known NP-complete problem that can be reduced to Hamiltonian Path.

For a given graph G we can solve Hamiltonian Path by nondeterministically choosing edges from G that are to be included in the path. Then we traverse the path and make sure that we visit each vertex exactly once. This obviously can be done in polynomial time, and hence, the problem belongs to NP.

Now we have to find an NP-complete problem that can be reduced to Hamiltonian Path. A closely related problem is the problem to determine whether a graph contains a Hamiltonian cycle, that is, a Hamiltonian path that begin and end in the same vertex. Moreover, we know that Hamiltonian Cycle is NP-complete, so we may try to reduce this problem to Hamiltonian Path.

Given a graph $G = \langle V, E \rangle$ we construct a graph G' such that G contains a Hamiltonian cycle if and only if G' contains a Hamiltonian path. This is done by choosing an arbitrary vertex u in G and adding a copy, u', of it together with all its edges. Then add vertices v and v' to the graph and connect v with u and v' with u'; see Figure 1 for an example.



Figur 1: A graph G and the hamiltonian path reduced graph G'.

Suppose first that G contains a Hamiltonian cycle. Then we get a Hamiltonian path in G' if we start in v, follow the cycle that we got from G back to u' instead of u and finally end in v'. For example, consider the left graph, G, in Figure 1 which contains the Hamiltonian cycle $1, 2, 5, 6, 4, 3, 1$. In G' this corresponds to the path $v, 1, 2, 5, 6, 4, 3, 1', v'$.

Conversely, suppose G' contains a Hamiltonian path. In that case, the path must necessarily have endpoints in v and v'. This path can be transformed to a cycle in G. Namely, if we disregard v and v', the path must have endpoints in u and u' and if we remove u' we get a cycle in G if we close the path back to u instead of u'.

The construction won't work when G is a single edge, so this has to be taken care of as a special case. Hence, we have shown that G contains a Hamiltonian cycle if and only if G' contains a Hamiltonian path, which concludes the proof that Hamiltonian Path is NP-complete. $\square$

Spanning trees with restricted degrees Show that the following problem is NP-complete: Given an undirected graph $G = (V, E)$ and an integer k, determine if G contains a spanning tree T such that each vertex of the tree has maximum degree k.

Solution to Spanning trees with restricted degrees
First note that the problem can be solved by the following nondeterministic algorithm:

1. For each edge in E, choose nondeterministically if it is to be included in T.
2. Check that T is a tree and that each vertex has degree less than k.

This means that the problem is in NP. Now we need to reduce a problem known to be NP-complete to our spanning tree problem. In this way we can state that determining whether a graph has a k-spanning tree is at least as hard as every other problem in NP.

Consider the problem HAMILTONIAN PATH that was shown to be NP-complete in the previous exercise. Can we reduce this problem to the spanning tree problem? That is, can we solve HAMILTONIAN PATH if we know how to solve the spanning tree problem? We claim that we can and that G has a Hamiltonian path if and only if it has a spanning tree with vertex degree ≤ 2.

It is easy to see that such a spanning tree is a Hamiltonian path. Since it has degree ≤ 2 it cannot branch and since it is spanning only two vertices can have degree < 2. So the spanning tree is a Hamiltonian path. If, on the other hand, G contains a Hamiltonian path, this path must be a spanning tree since the path visits every node and a path trivially is a tree.

We have reduced HAMILTONIAN PATH to the spanning tree problem and, therefore, our problem is NP-complete. $\square$

Polynomial reduction Construct a polynomial reduction from 3CNF-SAT to EQ-GF[2], satisfiability problem for a system of polynomial equations over GF[2] (ie, integers modulo 2).

Solution to Polynimoal reduction
Suppose we have a formula $\varphi \in$ 3CNF-SAT, for example

$$\varphi(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee x_3 \vee x_4)$$

Reducing 3CNF-SAT to EQ-GF[2] means that, for every formula φ, we can create an equation that is solvable if and only if φ is satisfiable. Now, φ is only satisfiable if every clause can be satisfied simultaneously, so let's start with finding an equation for an arbitrary clause $(x \vee y \vee z)$, where x, y and z are considered as literals rather than variables. For a start, the clause is satisfiable with one of the literals set to true, so in that case the equation $x + y + z = 1$ is also solved. But this fails if two literals are true! We can fix this by adding three more terms, where one and only one is 1 if two variables are 1. We get

$$x + y + z + xy + yz + xz = 1$$

Again, this is not complete. if all variables are set to 1, it fails. Of course, the solution is to add a term for this case, xyz. Our result is

$$x + y + z + xy + yz + xz + xyz = 1$$

If one of the literals in the clause happened to be inverted, e.g. we have $\overline{x}$ instead of x, replace x by $(1 + x)$ in the equation. The first clause in the example would be turned into

$$x_1 + x_2 + (1 + x_3) + x_1x_2 + x_1(1 + x_3) + x_2(1 + x_3) + x_1x_2(1 + x_3) = 1$$

We are now ready to create an equation from an arbitrary 3-CNF formula φ. For every clause φ_i in φ, create a corresponding equation Q_i. Our claim is that this system of equations is solvable if and only if φ is satisfiable and we must prove that this is correct.

To begin with, we note that $(1+x)$ is a proper way to handle inverses, $1+1=0$ and $1+0=1$, so in the following we will only consider literal values and not variables.

1. ($\Leftarrow$) If an equation in Q_i is satisfiable, then there exists an assignment to the variables such that each left hand expression is summed to 1. Hence, at least one of the literals is 1 and the corresponding literal in the boolean formula set true would make it's clause satisfied. The equation system Q_i being satisfied means that each equation is satisfied and consequently is each clause in φ satisfied.
2. ($\Rightarrow$) When φ is satisfied, we have an assignment on the variables such that each clause is satisfied. There are three cases for the clauses, (i) one literal is true, (ii) two literals are true and (iii) three literals are true. By construction, the corresponding equations are all satisfied.

□

Is an Euler graph k-colourable? Many problems of the type *determine whether the graph G has the property e* can be simplified if we assume that the graph has any particular characteristic. We will study a special case. We say that a connected graph is an *Euler graph* if each vertex of the graph has even degree. We now want to determine whether a graph is k-colourable. We have more specifically the following problem:

INPUT: An Euler graph G and an integer k.
OUTPUT: **YES** if the graph is k-colourable. **NO** otherwise.

For $k \leq 2$, there is a polynomial algorithm to determine coloring. Therefore, we assume that $k \geq 3$. Show if there is a polynomial algorithm to solve the above problem, or if the problem is NP-complete.

Solution to Is an Euler graph k-colourable?
The problem is NP-complete. It is easy to show that the problem is in NP by guessing a coloring and verify that it is correct. To show completeness, we reduce the general k-coloring problem to our problem. We assume that $k \geq 3$. Suppose we have a graph G. We reduce it to an Euler graph G' as follows: There must be an even number of vertices with odd degree. Divide the vertices of the pair. Each pair introduces a new vertex. Add two edges to this vertex, one from each old vertex that generated the new one. The graph G plus the new corners and edges represent the graph G'. It is easy to see that G' is an Euler graph.

We now show that G is k-colourable $\Leftrightarrow$ G' is k-colourable. Suppose that f is a coloring of G, it means that each vertex x has a color $f(x)$. We define a k-coloring f' of G' as follows: If x is a vertex of both G' and G then $f'(x) = f(x)$. If x does not belong to G, there are two adjecent vertices y and z in G. We set then $f(x)$ to any available color other than $f(y)$ and $f(z)$. (It is possible to do so since $k \geq 3$.) The implication in the other direction is trivial.

□
