

# Lab11-NP Reduction

CS214-Algorithm and Complexity, Xiaofeng Gao & Lei Wang, Spring 2021.

\* If there is any problem, please contact TA Yihao Xie.

\* Name: [Xin Xu](#) Student ID: [519021910726](#) Email: [xuxin20010203@sjtu.edu.cn](mailto:xuxin20010203@sjtu.edu.cn)

1. We are feeling experimental and want to create a new dish. There are various ingredients we can choose from and we'd like to use as many of them as possible, but some ingredients don't go well with others. If there are  $n$  possible ingredients (numbered 1 to  $n$ ), we write down an  $n \cdot n$  matrix giving the discord between any pair of ingredients. This discord is a real number between 0.0 and 1.0, where 0.0 means "they go together perfectly" and 1.0 means "they really don't go together." Here's an example matrix when there are five possible ingredients.

	1	2	3	4	5
1	0.0	0.4	0.2	0.9	1.0
2	0.4	0.0	0.1	1.0	0.2
3	0.2	0.1	0.0	0.8	0.5
4	0.9	1.0	0.8	0.0	0.2
5	1.0	0.2	0.5	0.2	0.0

In this case, ingredients 2 and 3 go together pretty well whereas 1 and 5 clash badly. Notice that this matrix is necessarily symmetric; and that the diagonal entries are always 0.0. Any set of ingredients incurs a penalty which is the sum of all discord values between pairs of ingredients. For instance, the set of ingredients (1, 3, 5) incurs a penalty of  $0.2 + 1.0 + 0.5 = 1.7$ . We define the EXPERIMENTAL CUISINE as follows:

Given  $n$  ingredients to choose from, the  $n \times n$  discord matrix and integer  $k$  and a number  $p$ , decide whether there exists a collection of at least  $k$  ingredients that has a penalty  $\leq p$

Prove that  $3\text{-SAT} \leq_p \text{EXPERIMENTAL CUISINE}$

**Proof.** Because  $3\text{-SAT} \leq_p \text{INDEPENDENT SET}$ , so we will prove that  $\text{INDEPENDENT SET} \leq_p \text{EXPERIMENTAL CUISINE}$ . We will construct an independent set to an experimental cuisine. For every two vertices  $v_i, v_j$  in graph  $G(V, E)$ , if  $v_i$  and  $v_j$  are not connected, the discord number is 0. And if  $v_i$  and  $v_j$  are connected, the discord number is 1. Assign  $p = 0$ , so that we can only choose ingredients with discord number is 0, which equals to find out the independent set of the original graph  $G$ . So,  $\text{INDEPENDENT SET} \leq_p \text{EXPERIMENTAL CUISINE}$ . And because of transitivity,  $3\text{-SAT} \leq_p \text{EXPERIMENTAL CUISINE}$ .  $\square$

2. An induced subgraph  $G' = (V', E')$  of a graph  $G = (V, E)$  is a graph that satisfies  $V' \subseteq V$  and  $E' = \{(u, v) \in E | u, v \in V'\}$ . Given two graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  and an integer  $b$ , we need to decide whether  $G_1$  and  $G_2$  have a common induced subgraph  $G_c$  with at least  $b$  nodes. This problem is called MAXIMUM COMMON SUBGRAPH (MCS). Prove that MCS is NP-complete. (Hint: reduce from INDEPENDENT-SET)

**Proof.** Prove MCS a NP Problem:

For a certification to determine whether or not the given subgraph is the maximum common subgraph, we should traverse all the nodes in the given graph to compare with  $G_1$  and  $G_2$  to ensure its existence. Finally we should check the number of nodes of given graph equals to  $b$  or not. It's easy to know the traverse and check time are in polynomial time. So, MCS is a NP Problem.

Prove  $\text{INDEPENDENT SET} \leq_p \text{MCS}$ :

We will construct a MCS from an independent set problem. Define  $G_1(V_1, E_1)$  a random graph and  $G_2(V_2, E_2)$  a complete graph, with  $|V_2| = |V_1|$ . And there are two complement graphs of them,  $G'_1$  and  $G'_2$ . For  $G'_1$  and  $G'_2$ , we find out the independent set of them. The independent set of complement graph is corresponding to the complete graph of the original graph. And we define the induced graph a complete graph, so the independent set of complement graph can reduce to MCS. Because INDEPENDENT SET is a NP Complete Problem, MCS is a NP Complete Problem too.  $\square$

3. Let us define the  $k$ -spanning tree as a spanning tree in which each node has a degree  $\leq k$ . Given a graph  $G = (V, E)$  and a positive integer  $k$ , we need to decide whether there exists a  $k$ -spanning tree in  $G$ . Prove that this problem is NP-complete. (Hint: reduce from HAMILTONIAN-CYCLE)

**Proof.** Prove  $k$ -spanning tree a NP Problem:

For the certification of this problem, we should traverse all the vertex of the given spanning tree to determine whether the degree  $\leq k$  and the set of given spanning tree nodes equals to the original vertex set. It's easy to know the traverse and check time are in polynomial time. So,  $k$ -spanning tree is a NP Problem.

Prove HAMILTONIAN CYCLE  $\leq_p$  K-SPANNING TREE:

It's clear that when  $k = 2$ ,  $k$ -spanning tree problem is equal to Hamiltonian Cycle problem. So we will extend it. For a random graph  $G(V, E)$ , we will add  $k - 2$  new nodes to every old node in  $G$ . So, if the old graph has a hamiltonian cycle, the new graph has a  $k$ -spanning tree. And if we can find a  $k$ -spanning tree of the new graph, we can just remove the new-added nodes to get a hamiltonian cycle. So, HAMILTONIAN CYCLE  $\leq_p$  K-SPANNING TREE. And because HAMILTONIAN CYCLE is a NP Complete Problem, K-SPANNING TREE is a NP Complete Problem too.  $\square$

4. We define the decision problem of KNAPSACK PROBLEM as follows:

Given  $n$  indivisible objects, each with a weight of  $w_i > 0$  kilograms and a value  $v_i > 0$ , a knapsack with capacity of  $W$  kilograms and a number  $k$ , decide whether there is a collection of objects that can be put into the knapsack with a total value  $V \geq k$ .

Prove that KNAPSACK PROBLEM is NP-complete.

**Proof.** Prove KNAPSACK PROBLEM a NP Problem:

According to the dynamic programming, the running time of KNAPSACK PROBLEM is  $O(nW)$ . Because the length of input is  $\log n$ , so the running time is exponential in the input size. As a result, KNAPSACK PROBLEM a NP Problem.

Prove SUBSET SUM  $\leq_p$  KNAPSACK PROBLEM:

For a SUBSET SUM problem, we will construct it into a KNAPSACK PROBLEM. Define that  $k = W, v_i = w_i$ , and the limit conditions of the newly constructed KNAPSACK PROBLEM become  $\sum_{i \in S} w_i \leq W, \sum_{i \in S} v_i \geq k = W, w_i = v_i, \forall i \in N$ , which means this KNAPSACK PROBLEM satisfies if and only if  $\sum_{i \in S} w_i = W$ . So, SUBSET SUM  $\leq_p$  KNAPSACK PROBLEM. And because SUBSET SUM is a NP Complete Problem, KNAPSACK PROBLEM is a NP Complete Problem too.  $\square$

**Remark:** Please include your .pdf, .tex files for uploading with standard file names.