

# Lab11-NP Reduction

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

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

\* Name: WendiChen Student ID: 519021910071 Email: chenwendi-andy@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  $3\text{-SAT} \leq_p \text{EXPERIMENTAL CUISINE}$

## Solution.

In the class, we have already proved that  $3\text{-SAT} \leq_p \text{INDEPENDENT-SET}$ . Thus, we can transform the original problem into proving

$$\text{INDEPENDENT-SET} \leq_p \text{EXPERIMENTAL CUISINE}$$

Given an instance  $(G, k)$  of INDEPENDENT-SET, we construct an instance  $(D, k, 0)$  of EXPERIMENTAL CUISINE that has a collection of at least  $k$  ingredients that has a penalty  $\leq 0$  iff  $(G, k)$  has an independent set of at least  $k$  vertices.

**Construction.** For each vertex in  $G$ , we add an ingredient to  $D$ . If there is any undirected edge between two vertices in  $G$ , then the discord between the two corresponding ingredients is 1. Otherwise, the discord is 0. Besides, we set  $p = 0$ . This procedure has a polynomial time complexity.

**Claim.**  $D$  has a collection of at least  $k$  ingredients that has a penalty  $\leq 0$  iff  $G$  has an independent set of at least  $k$  vertices.

## Proof.

$\Rightarrow$  Because the penalty is non-negative, the penalty should be 0. If the penalty is 0, then the discord between any two ingredients in the collection should be 0. Then, the corresponding vertices in  $G$  constructs an independent set and the number of vertices is at least  $k$ .

$\Leftarrow$  Because there is no edge between any two vertices in the independent set. Then, the discord between any two corresponding ingredients should be 0. Thus, the corresponding ingredients form a collection that has a penalty  $\leq 0$  and the number of ingredients is at least  $k$ .

By using the *transitivity*, we have proved that  $3\text{-SAT} \leq_p \text{EXPERIMENTAL CUISINE}$ .  $\square$

2. An induced subgraph  $G = (V', E')$  of a graph  $G = (V, E)$  is a graph that satisfies  $V' \subseteq V$  and  $E' = \{(u, v) \in E \mid u, v \in V'\}$ . 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)

**Solution.**

Because INDEPENDENT-SET is NP-complete, we can transform the original problem into proving

$$\text{INDEPENDENT-SET} \leq_p \text{MAXIMUM COMMON SUBGRAPH}$$

Given an instance  $(G, k)$  of INDEPENDENT-SET, we construct an instance  $(G_1, G_2, k)$  of MAXIMUM COMMON SUBGRAPH where  $G_1$  and  $G_2$  have a common induced subgraph  $G_c$  with at least  $k$  vertices iff  $(G, k)$  has an independent set of at least  $k$  vertices.

**Construction.** First, we construct the complementary graph of  $G$  and denote it as  $G_1$ . Then, we construct the graph  $G_2$  which is a complete graph consisting of  $k$  vertices. This procedure has a polynomial time complexity.

**Claim.**  $G_1$  and  $G_2$  have common induced subgraph  $G_c$  with at least  $k$  vertices iff  $G$  has an independent set of at least  $k$  vertices.

**Proof.**

$\Rightarrow$   $G_2$  is a complete graph consisting of  $k$  vertices, hence there should be a subgraph  $G_c$  of  $G_1$  that is a complete graph of  $k$  vertices. Because  $G_1$  is the complementary graph of  $G$ , these  $k$  vertices must construct an independent set in  $G$ . Otherwise, there will be edges between these  $k$  vertices in  $G$  and these edges are not in  $G_1$ , thus these  $k$  vertices can not construct a complete graph in  $G_1$ .

$\Leftarrow$  Because there is no edge in  $G$  between any two vertices in the independent set. Thus, these vertices in the independent set must construct a complete subgraph in  $G_1$ . And a complete graph consisting of more than  $k$  vertices must contain a subgraph of  $k$  vertices which is also a complete graph. Therefore, this subgraph is the common induced subgraph  $G_c$  of  $G_1$  and  $G_2$  with  $k$  vertices.

Thus,  $\text{INDEPENDENT-SET} \leq_p \text{MAXIMUM COMMON SUBGRAPH}$  and MAXIMUM COMMON SUBGRAPH is NP-complete.  $\square$

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)

**Solution.**

Because HAMILTONIAN-CYCLE is NP-complete, we can transform the original problem into proving

$$\text{HAMILTONIAN-CYCLE} \leq_p \text{K-SPANNING TREE}$$

Given an instance  $G = (V, E)$  of HAMILTONIAN-CYCLE, we construct a set of instances  $\{(G_1, k), \dots, (G_{|E|}, k)\}$  of K-SPANNING TREE that there exist  $(G_i, k)$  which has a  $k$ -spanning tree iff  $G$  has a Hamiltonian cycle.

**Construction.** For  $G_i$ , we choose a distinct edge  $e_i = (v_{s_i}, v_{t_i})$ . Then, we add two assisting vertices  $v_{p_i}$  and  $v_{q_i}$ .  $G_i$  is constructed by letting  $G_i = (V + \{v_{p_i}, v_{q_i}\}, E + \{(v_{p_i}, v_{s_i}), (v_{q_i}, v_{t_i})\} - \{(v_{s_i}, v_{t_i})\})$ . The procedure of constructing all  $G_i$  has a polynomial time complexity.

**Claim.** There exists  $G_i$  that has a 2-spanning tree iff  $G$  has a Hamiltonian cycle.

**Proof.**

$\Rightarrow$   $G_i$  has a 2-spanning tree means that it contains all the vertices in  $G_i$  and each vertex has a degree  $\leq 2$ . So, the spanning tree is reduced to a “linked list”. Because the degree of  $v_{p_i}$  and  $v_{q_i}$  is 1, the linked list must have the following pattern.

$$v_{p_i} - v_{s_i} - \cdots - v_{t_i} - v_{q_i}$$

Remove vertexes  $v_{p_i}, v_{q_i}$ , edges  $(v_{p_i}, v_{s_i}), (v_{q_i}, v_{t_i})$  and add edge  $(v_{s_i}, v_{t_i})$ , then the cycle contains all the vertices in  $G$  we get a Hamiltonian cycle in  $G$ .

$\Leftarrow$  If there is a Hamiltonian cycle in  $G$ , we can remove an arbitrary edge  $(v_{s_i}, v_{t_i})$  from the cycle and the cycle becomes a linked list. Because  $v_{s_i}, v_{t_i}$  are the head and tail of the linked list and all the vertexes in  $G$  are contained in the cycle, we can add vertexes  $v_{p_i}, v_{q_i}$  and edges  $(v_{p_i}, v_{s_i}), (v_{q_i}, v_{t_i})$  to the linked list and get a 2-spanning tree of  $G_i$ .

Thus, HAMILTONIAN-CYCLE  $\leq_p$  K-SPANNING TREE and K-SPANNING TREE is NP-complete.  $\square$

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.

**Solution.**

Because SUBSET SUM is NP-complete, we can transform the original problem into proving

$$\text{SUBSET SUM} \leq_p \text{KNAPSACK PROBLEM}$$

Given an instance  $(S, W)$  of SUBSET SUM, we construct an instance  $(S', W', k')$  of KNAPSACK PROBLEM that there exists a collection of objects  $\in S'$  that can be put into the knapsack with a total value  $V \geq k'$  iff there is a subset of  $S$  that adds up to exactly  $W$ .

**Construction.** For each natural number  $w_i \in S$ , we construct a corresponding object  $\in S'$  whose weight  $w'_i = w_i$  and value  $v'_i = w_i$ . And we set  $W' = W$  and  $k' = W$ . This procedure has a polynomial time complexity.

**Claim.** There exists a collection of objects  $\in S'$  that can be put into the knapsack with a total value  $V \geq k'$  iff there is a subset of  $S$  that adds up to exactly  $W$ .

**Proof.**

$\Rightarrow$  We denote the collection of objects that we choose is  $S_c$ . Then we have  $\sum_{s_i \in S_c} w'_i \leq W'$  and  $\sum_{s_i \in S_c} v'_i \geq k'$ . According to the construction, we can derive that  $\sum_{s_i \in S_c} w_i = W$ . Thus, we can choose the corresponding natural numbers  $\in S$  and get a subset of  $S$  that adds up to exactly  $W$ .

$\Leftarrow$  For each natural number in the subset of  $S$ , we can choose the corresponding objects  $\in S'$ . And we get  $\sum_{s_i \in S_c} w'_i = W \leq W'$  and  $\sum_{s_i \in S_c} v'_i = W \geq k'$ . Thus, there exists a collection of objects  $\in S'$  that can be put into the knapsack with a total value  $V \geq k'$ .

Thus, SUBSET SUM  $\leq_p$  KNAPSACK PROBLEM and KNAPSACK PROBLEM is NP-complete.  $\square$

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