Lab11-NP Reduction

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

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

∗ Name:Yanjie Ze   Student ID: 519021910706   Email: zeyanjie@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 $\leqslant p$

Prove that 3-SAT $\leq_p$ Experimental Cuisine

Proof.
Basic idea: By proving that INDEPENDENT-SET $\leq_p$ EXPERIMENTAL CUISINE, we prove that 3-SAT $\leq$ EXPERIMENTAL CUISINE.

We construct a graph G. Each vertex v in G represents an ingredient.

For the edges of G:

- If the discord between the ingredient i and the ingredient j is 0, the edge $e(i, j)$ between the vertex v_i and the vertex v_j does not exist.
- If the discord between the ingredient i and the ingredient j is not 0, the edge $e(i, j)$ between the vertex v_i and the vertex v_j exists.

Claim: *there exists an independent set of size at least k iff there exists an experimental cuisine collection of at least k with a penalty=0.*

⇒ If in G there exists an independent set of size at least k, then there must exists an experimental cuisine collection of at least k with a penalty $p = 0$. Since each vertex in G represents an ingredient, we use the set consisting of corresponding ingredients of the vertices in the independent set as the cuisine.

⇐ If there exists a cuisine collection of size at least k with a penalty $p = 0$, we can use the corresponding vertices of the ingredients to construct the independent set, with size of at least k.

Therefore, we prove that INDEPENDENT-SET $\leq_p$ EXPERIMENTAL CUISINE.

By 3-SAT $\leq_p$ INDEPENDENT-SET and INDEPENDENT-SET $\leq_p$ EXPERIMENTAL CUISINE, we have:

$$\text{3-SAT} \leq_p \text{EXPERIMENTAL CUISINE.}$$

□

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)

   Solution.
   We construct two graphs $G_1 = (V, E)$ and $G_2 = (V, \emptyset)$.

   We prove that given a graph G_1, G_1 has the independent set of size at least b if and only if G_1 and G_2 have a common induced subgraph G_c with at least b nodes.

   Claim: G_1 has the independent set of size at least b iff G_1 and G_2 have a common induced subgraph G_c with at least b nodes.

   Proof:
   $\Rightarrow$ G_1 has the independent set V_{ind} of size at least b. Then the maximum common subgraph G_c of G_1 and G_2 has exactly the same vertices as V_{ind}, and no edges are in G_c. Therefore, G_c has at least b nodes.

   $\Leftarrow$ The maximum subgraph G_c of G_1 and G_2 has at least b nodes. Then the vertices of G_c are the same as the vertices in the independent set V_{ind}. Therefore, V_{ind} has the size of at least b.

   Therefore, INDEPENDENT-SET $\leq_p$ MCS, and MSC is **NP-complete**.

   □

3. Let us define the k-spanning tree as a spanning tree in which each node has a degree $\leqslant k$. 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.
   Basic idea: We prove in the following order:

   (a) HAMILTONIAN-CYCLE $\leq_p$ HAMILTONIAN-PATH
   (b) HAMILTONIAN-PATH $\leq_p$ K-SPANNING TREE
   (c) HAMILTONIAN-CYCLE $\leq_p$ K-SPANNING TREE

   Proof:

   (a) HAMILTONIAN-CYCLE $\leq_p$ HAMILTONIAN-PATH
   Given a graph $G = (V, E)$, we construct a new graph G' such that G contains a HAMILTONIAN-CYCLE if and only if G' contains a HAMILTONIAN-PATH.
   Construct G':

i. Copy all the edges and all the vertices of G in G', as shown in Fig. 1.

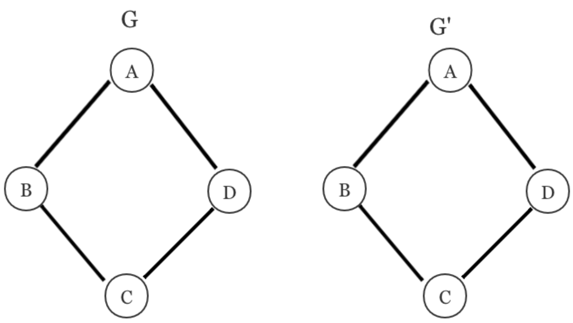

Figure 1: Construct G', step 1

ii. Select a random vertex v in G'. Create another new vertex named v' and connect v' to all the vertices that v is connected to, as shown in Fig. 2.

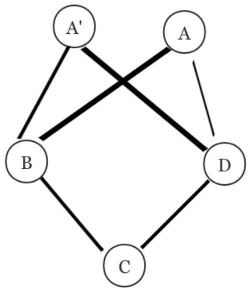

Figure 2: Construct G', step 2

iii. Add two more vertices as the start point and the end point. Connect the start point with v' and the end point with v, as shown in Fig. 3.

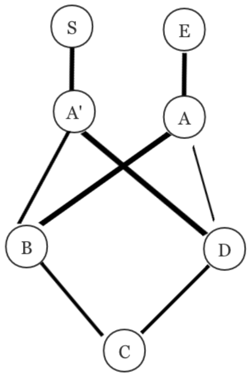

Figure 3: Construct G', step 3

If G contains a HAMILTONIAN-CYCLE, there exists a HAMILTONIAN-PATH in G', starting from the start point and ending at the end point.
For the example we construct, the HAMILTONIAN-CYCLE in G is

$$\{(A, B), (B, C), (C, D), (D, A)\}$$

Then in G', the HAMILTONIAN-PATH is

$$\{(S, A'), (A', B), (B, C), (C, D), (D, A), (A, E)\}$$

If G' contains a HAMILTONIAN-PATH, we can transform this path in G' into the HAMILTONIAN-CYCLE in G. This is because the HAMILTONIAN-PATH in G' must pass the start point and the end point. We delete them and make v' back to v, which will transform the HAMILTONIAN-PATH in G' into the HAMILTONIAN-CYCLE in G.

For this example, in G', the HAMILTONIAN-PATH is

$$\{(S, A'), (A', B), (B, C), (C, D), (D, A), (A, E)\}$$

Delete S, E, A' and transform (A', B) into (A, B). Then G' becomes G and HAMILTONIAN-PATH becomes HAMILTONIAN-CYCLE:

$$\{(A, B), (B, C), (C, D), (D, A)\}$$

Thus, we prove that HAMILTONIAN-CYCLE $\leq_p$ HAMILTONIAN-PATH.

(b) HAMILTONIAN-PATH $\leq_p$ K-SPANNING TREE

Given a graph G, G contains a HAMILTONIAN-PATH iff G contains a k-spanning tree ,$k = 2$.

This is easy to prove, since a HAMILTONIAN-PATH is exactly a special case of the k-spanning tree. When $k = 2$, the nodes either has the degree 1 or degree 2, which means this tree is also a path.

Therefore, HAMILTONIAN-PATH $\leq_p$ K-SPANNING TREE.

(c) HAMILTONIAN-CYCLE $\leq_p$ K-SPANNING TREE

From (a) and (b) we know that: HAMILTONIAN-CYCLE $\leq_p$ HAMILTONIAN-PATH, HAMILTONIAN-PATH $\leq_p$ K-SPANNING TREE

Therefore, HAMILTONIAN-CYCLE $\leq_p$ K-SPANNING TREE.

Since HAMILTONIAN-CYCLE is NP-Complete, K-SPANNING TREE is also NP-Complete.

□

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 $V \geqslant k$.

   Prove that Knapsack Problem is NP-complete.

Solution.

Basic idea: We can prove that SUBSET-SUM $\leq_p$ KNAPSACK PROBLEM.

SUBSET-SUM: Given a set S of non-negative integers ,$S = \{s_1, s_2, ..., s_n\}$, does there exist a subset of S with the sum of its elements equal to T?

Reduction: Given a set $S = \{s_1, s_2, ..., s_n\}$ consisting of non-negative numbers, We construct a knapsack with capacity $W = T$, and set the value threshold $k = T$.

We set n indivisible objects, each with a weight of $w_i = s_i$.

Claim: The set S has a subset with the sum of elements equal to T iff there exists a collection of objects that can be put into the knapsack with a total value $V = T$.

$\Rightarrow$ If the set S has a subset with the sum of elements equal to T, the element of this subset will correspond to one object, $w_i = s_i$. Then we place these corresponding objects in the knapsack, which will have a total value of T.

$\Leftarrow$ If there exists a collection of objects that can be put into the knapsack with a total value $V = T$, we construct the subset S' of S. The elements of S' are the elements corresponding to the objects in the knapsack, $s_i = w_i$. Therefore, the sum of elements in S' is equal to T.

Thus, SUBSET-SUM $\leq_p$ KNAPSACK PROBLEM.

Since SUBSET-SUM is NP-Complete, KNAPSACK PROBLEM is also NP-Complete.

$\square$

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