Algorithm Design and Analysis
Assignment 6
Deadline: Jun 20, 2024

Choose **two** questions to solve.

1. (50 points) Given an undirected graph $G = (V, E)$ and an integer k, decide if G has a spanning tree with maximum degree at most k. Prove that this problem is NP-complete.

Answer:

To show that the problem of deciding if a given undirected graph $G = (V, E)$ has a spanning tree with maximum degree at most k is NP-complete, we need to do two things:

1. Prove that the problem is in NP.

2. Prove that the problem is NP-hard by reducing a known NP-complete problem to it.

1. The Problem is in NP

To prove that the problem is in NP, we need to show that given a certificate (in this case, a spanning tree of G with maximum degree at most k), we can verify in polynomial time that the certificate is valid.

Certificate: A spanning tree T of G where the maximum degree of any vertex in T is at most k.

Verification: We can verify this by:

1. Checking that T is a subgraph of G.

2. Checking that T includes all vertices of G and that T is connected.

3. Checking that T has exactly $|V| - 1$ edges (which is the property of a tree).

4. Checking that the degree of every vertex in T is at most k.

Each of these checks can be performed in polynomial time, so the problem is in NP.

2. The Problem is NP-Hard

To prove that the problem is NP-hard, we need to show a polynomial-time reduction from a known NP-complete problem to our problem. We will reduce from the **Hamiltonian Path problem**, which is known to be NP-complete.

Reduction

1. **Input:** A graph $G' = (V', E')$ for the Hamiltonian Path problem.

2. **Transformation:**

- Construct a new graph G from G' by adding a new vertex v_0 and connecting v_0 to every vertex in V'. Let $G = (V, E)$, where $V = V' \cup \{v_0\}$ and $E = E' \cup \{(v_0, v) \mid v \in V'\}$.

- Set $k = 2$.

3. **Correctness:**

Suppose G' has a Hamiltonian Path. Then there exists a path P in G' visiting each vertex exactly once. We can construct a spanning tree T in G as follows:

- Include the edges of P in T.

- Add the edge from v_0 to one end of P.

In this tree T, the degree of v_0 is 1, and the degree of every other vertex is at most 2 (since they were part of a Hamiltonian Path and each internal vertex in the path has degree 2, while the endpoints have degree 1). Hence, the maximum degree in T is 2.

Conversely, suppose G has a spanning tree T with maximum degree at most 2. Since T is a spanning tree, T must include v_0. The degree of v_0 must be 1 (since the maximum degree is 2 and v_0 connects to all other vertices). Removing v_0 and its incident edge from T leaves a path in G' that visits all vertices exactly once (a Hamiltonian Path).

Since the transformation can be performed in polynomial time and the Hamiltonian Path problem is NP-complete, the problem of determining if G has a spanning tree with maximum degree at most k is NP-hard.

2. (50 points) An *0-1 integer linear program* is similar to a linear program, except that each variable x_i is required to be either 0 or 1.

$$\begin{aligned}
\text{maximize} \quad & c_1x_1 + \cdots + c_nx_n \\
\text{Subject to} \quad & a_{11}x_1 + \cdots + a_{1n}x_n \le b_1 \\
& a_{21}x_1 + \cdots + a_{2n}x_n \le b_2 \\
& \quad \vdots \\
& a_{m1}x_1 + \cdots + a_{mn}x_n \le b_m \\
& x_1, \ldots, x_n \in \{0, 1\}
\end{aligned}$$

We have seen in the class that a linear program can be solved in polynomial time. However, we will see in this question that this is unlikely for 0-1 integer linear programs.

(a) Prove that, for a given input k, deciding if there is a feasible solution $(x_1, \ldots, x_n)$ such that $c_1x_1 + \cdots + c_nx_n \ge k$ is NP-complete.

(b) Prove that it is NP-complete to even decide if there is a feasible solution.

Answer:

Part (a):

1. The Problem is in NP:

Certificate: A vector $x = (x_1, x_2, \ldots, x_n)$ where each $x_i \in \{0, 1\}$.

Verification: Given x:

- Check if $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$ for each constraint.

- Check if $c_1x_1 + c_2x_2 + \cdots + c_nx_n \geq k$.

These checks can be performed in polynomial time, so the problem is in NP.

2. The Problem is NP-hard:

We will reduce from the **Subset Sum** problem, which is known to be NP-complete.

3. Reduction from Subset Sum to 0-1 integer linear program:

Given an instance of the Subset Sum problem, construct the following 0-1 ILP:

Objective function: Maximize $\sum_{i=1}^{n} a_i x_i$.

Constraint: $\sum_{i=1}^{n} a_i x_i \leq S$.

Set $k = S$.

The resulting 0-1 ILP is:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} a_i x_i \\ \text{subject to} \quad & \sum_{i=1}^{n} a_i x_i \leq S \\ & x_i \in \{0, 1\} \quad \forall i \end{aligned}$$

If there is a subset of $\{a_1, a_2, \ldots, a_n\}$ that sums to S, then there is a feasible solution to the 0-1 ILP where the objective value is at least S. Conversely, if the 0-1 ILP has a feasible solution with the objective value at least S, then there is a subset of the given integers that sums to S. This shows the equivalence of the two problems and thus proves that the 0-1 ILP problem is NP-hard.

Since the problem is in NP and NP-hard, it is NP-complete.

Part (b):

1. The Problem is in NP:

Certificate: A vector $x = (x_1, x_2, \ldots, x_n)$ where each $x_i \in \{0, 1\}$.

Verification: Given x:

- Check if $a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \le b_1$ for each constraint.

These checks can be performed in polynomial time, so the problem is in NP.

The Problem is NP-hard:

We will reduce from the **3-SAT** problem, which is known to be NP-complete.

3. Reduction from 3-SAT to 0-1 ILP:

Given an instance of 3-SAT, construct the following 0-1 ILP:

For each variable x_i in the 3-SAT instance, introduce a corresponding variable $x_i \in \{0, 1\}$ in the ILP.For each clause in 3-SAT instance, construct a constraint in ILP such that the constraint is satisfied if and only if at least one of the literals in the clause is true.

For example, consider a clause $(x_1 \vee \neg x_2 \vee x_3)$. The corresponding ILP constraint could be:

$$x_1 + (1 - x_2) + x_3 \ge 1$$

This constraint ensures that at least one of x_1, $\neg x_2$, or x_3 is true (equivalent to $x_1 = 1$ or $x_2 = 0$ or $x_3 = 1$).

Repeat this for all clauses in the 3-SAT instance to form the complete ILP. The resulting ILP will have a feasible solution if and only if the original 3-SAT instance is satisfiable.

Since 3-SAT is NP-complete and we have shown a polynomial-time reduction to the feasibility problem of a 0-1 ILP, the feasibility problem of a 0-1 ILP is NP-hard.

Since the problem is in NP and NP-hard, it is NP-complete.

3. (50 points) Suppose we want to allocate n items $S = \{1, \ldots, n\}$ to two agents. The two agents may have different values for each item. Let $u_1, u_2, \ldots, u_n$ be agent 1's values for those n items, and $v_1, v_2, \ldots, v_n$ be agent 2's values for those n items. An allocation is a partition (A, B) for S, where A is the set of items allocated to agent 1 and B is the set of items allocated to agent 2. An allocation (A, B) is *envy-free* if, based on each agent's valuation, (s)he believes the set (s)he receives is (weakly) more valuable than the set received by the other agent. Formally, (A, B) is envy-free if

$$\sum_{i \in A} u_i \ge \sum_{j \in B} u_j \qquad \text{agent 1 thinks } A \text{ is more valuable}$$

and

$$\sum_{i \in B} v_i \ge \sum_{j \in A} v_j \qquad \text{agent 2 thinks } B \text{ is more valuable.}$$

Prove that deciding if an envy-free allocation exists is NP-complete.

4. (50 points) In the *k-means* problem, you are given a set of data points $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d\}$ and a positive integer k as inputs, and you need to output k "centers" $\mathbf{c}_1, \dots, \mathbf{c}_k \in \mathbb{R}^d$ and a k-partition $(C_1, \dots, C_k)$ of the data points D such that the data points in C_i is assigned to the center $\mathbf{c}_i$. The objective is to minimize the following value

$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c}_i\|^2$$

which is the sum of the squared distances from the data points to their assigned centers.

Prove that the following problem is NP-complete: given a k-means instance (D, k) and a non-negative value θ, decide if there exists a solution $((\mathbf{c}_1, \dots, \mathbf{c}_k), (C_1, \dots, C_k))$ that makes the objective value at most θ.

5. How long does it take you to finish the assignment (including thinking and discussion)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Please write down their names here.

   To prepare for the final exams, I choose two questions to answer.The NP problem contains so much fun that I will dive into more learning in the near future. Thanks very much for my teacher and teaching assistants' devotion to the class and reviewing my work.Best wishes!