

# Assignment 6

Kailing Wang 521030910356

January 7, 2023

**Problem 1.** (30 points) Given an undirected unweighted graph  $G = (V, E)$ , prove that it is NPcomplete to decide if there is a simple path with length exactly  $|V|/2$  (a simple path is a path that do not visit a vertex more than once).

**Answer 1.**

To prove this we should use reduction:

Basically we know that given a path, it's NP to check if it's legal. We can simply calculate the length and check the validity of edges and vertices one by one.

Recap the definition of Hamiltonian path: a simple path with length exactly  $|V|$ . Hamiltonian path is a NP-complete problem.

Then we have to prove  $\leq_P \text{HamiltonianPath}$  this problem:

Suppose we are to find a Hamiltonian path on  $G' = (V', E')$ ,  $|V'| = |V|/2$ . Expand  $G'$  by adding vertices but not adding edges in  $G'$  to get  $G'' = (V, E')$ . This way, we convert the Hamiltonian path problem into a  $|V|/2$  path problem. The Hamiltonian path problem is no stronger than the  $|V|/2$  path problem.

If we find the Hamiltonian path, this is the solution for the  $|V|/2$  path problem on  $G''$ . If we cannot find a Hamiltonian path, there must not be solution for the  $|V|/2$  path problem since the added vertices are separate.

Now we can say the original problem is NP-complete.

**Problem 2.** (30 points) Suppose we would like to complete  $n$  jobs. Each job  $i$  has a release time  $r_i \in \mathbb{Z}^+$  and a deadline  $d_i \in \mathbb{Z}^+$ , and it requires  $t_i \in \mathbb{Z}^+$  units of time to complete. Suppose we can only process one job at a time (i.e., we cannot simultaneously process two jobs), and a job must be done within a consecutive time interval (i.e., we cannot pause a job and resume it at a later time). Prove that it is NP-complete to decide if there is a schedule so that all the jobs are completed.

**Answer 2.**

First we know for ant given schedule, we only have to scan once to check if the timetable is legal and if all the jobs are finished. This is a NP problem.

Prove using reduction: this problem can be reduced by subset sum problem.

A subset sum problem is to determine if we can acquire a subset of  $S = \{a_1, a_2 \cdots a_n\}$  with sum  $k$ . Suppose we have to deal with  $n+1$  jobs. One of the job  $x$  is released at  $r_x = k + 1$  and the deadline is  $k + 1 + t_x$ , which means we have to start it at  $r_x$ . The other jobs have  $r_i = 1$  and deadline  $sum(S) + t_x$ . Whether there is a schedule is determined by whether there is a subset sum  $k$  in  $S$  because we can waste no time if we want all the job done.

If there is a such subset, we have a schedule by doing the concerning jobs in the subset in ant sequence, and then do the job  $x$ , and do the rest. If there isn't such subset, we must wast time before doing job  $x$ , and we waste some time. We can't finish all.

This problem is NP-complete.

**Problem 3.** (40 points) An integer linear program is similar to a linear program, except that each variable  $x_i$  is required to be an integer.

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

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 integer linear programs.

(a) (30 points) Prove that, for a given input  $k$ , deciding if there is a feasible solution  $(x_1, \dots, x_n)$  such that  $c_1x_1 + \cdots + c_nx_n \geq k$  is NP-complete.

(b) (10 points) Prove that it is NP-complete to even decide if there is a feasible solution.

**Answer 3.**

(a) According to the additional message, we only have to prove NP-hardness.

Try to use a vertex cover problem to reduce. Given a graph  $G = (V, E)$ , if  $x_i$  is connected with  $x_j$ , the constraint is  $x_i + x_j \geq 1$  for vertex cover and the LP for vertex cover is

$$\begin{aligned} & \min \sum_{v \in V} x_v \\ & x_v + x_u \geq 1, \forall uv \in E \\ & x_v \geq 0, \forall v \in V \\ & x_v \in \mathbb{Z}, \forall v \in V \end{aligned}$$

This is a LP in the form of our problem. So our problem is hard. Note that in (a) we can view  $c_1x_1 + \cdots + c_nx_n \geq k$  as a certain constraint.

If there is a such vertex cover, the cover is a solution for this LP. If there isn't, there couldn't be a feasible solution for we can't meet all the constraints.

(b) Remove the constraint  $c_1x_1 + \cdots + c_nx_n \geq k$ , the process is the same as (a).

**Problem 4.** (Bonus 60 points) [Ultimate Cake Cutting with Yuhao] Consider the cake cutting problem in Question 5 of Assignment 1. Recall that the objective is to allocate the cake to the  $n$  agents fairly such that each agent believes he receives the average value based on his preference. The formally model is recalled below.

The cake is modelled by the 1-dimensional interval  $[0,1]$ . Each agent  $i$  has a value density function  $f_i : [0,1] \rightarrow \mathbb{R}_{\geq 0}$  representing  $i$ 's preference over the cake  $[0,1]$ . Given an interval  $I \subseteq [0,1]$ , agent  $i$ 's value on  $I$  is given by the Riemann integral  $\int_I f_i(x)dx$ . An allocation is a collection of  $n$  intervals  $(I_1, \dots, I_n)$  such that every pair of intervals  $I_i$  and  $I_j$  can only intersect at the endpoints, where  $I_i$  is the interval allocated to agent  $i$ . An allocation  $(I_1, \dots, I_n)$  is proportional if  $\int_{I_i} f_i(x)dx \geq \frac{1}{n} \int_0^1 f_i(x)dx$  for each agent  $i$ , i.e., each agent  $i$  thinks he receives the average value based on his value density function.

In this question, we assume that each value density function  $f_i$  is piecewise-constant. That is, for each  $f_i$ , the cake  $[0,1]$  can be partitioned into finitely many intervals  $[0, x_1], [x_1, x_2], \dots, [x_{k-1}, x_k], [x_k, 1]$  such that  $f_i$  has a constant value on each of these intervals.

In Question 5 of Assignment 1, we have seen that a proportional allocation always exists and can be computed efficiently by Even-Paz algorithm. In this question, we will require just slightly more than proportionality.

Suppose Professor Yuhao Zhang is one of the  $n$  agents, and we would like to give him slightly more than his proportional value for his excellent service in AI2615. In particular, suppose Yuhao is agent 1. We would like to have an allocation  $(I_1, \dots, I_n)$  such that  $\int_{I_1} f_1(x)dx \geq \frac{2}{n} \int_0^1 f_1(x)dx$  for Yuhao, and  $\int_{I_i} f_i(x)dx \geq \frac{1}{n} \int_0^1 f_i(x)dx$  for each of the remaining agents  $i$ . We call such an allocation Yuhao-prioritized proportional.

Of course, Yuhao-prioritized proportional allocation may not exist. For example, if all the agents have the same value density function  $f_i(x) = 1$  on  $[0, 1]$ , the only proportional allocation is to allocate each agent an interval of length  $\frac{1}{n}$ , and Yuhao cannot be prioritized. In this question, we study the problem of deciding if Yuhao-prioritized proportional allocation exists for a given cake-cutting instance.

(a) (20 points) Suppose each agent is allowed to allocated multiple intervals. That is, each  $I_i$  can be a union of finitely many intervals in an allocation  $(I_1, \dots, I_n)$ . Show that deciding if Yuhao-prioritized proportional allocation exists is in P.

(b) (40 points) Consider the original setting where each  $I_i$  is required to be a connected interval. Prove that deciding if Yuhao-prioritized proportional allocation exists is NP-complete.

#### Answer 4.

(a) Finished orally to help others. It's easy to form the problem as a LP.

(b) It's very easy to check the validity of a given answer, and thus this problem is NP. Then we may use Maximum Weight Closed Subgraph problem? I guess.

**Problem 5.** (0 points) For practice and for fun, not for credits.

(a) Given an undirected graph  $G = (V, E)$  with  $n = |V|$ , decide if  $G$  contains a clique with size exactly  $n/2$ . Prove that this problem is NP-complete.

(b) Consider the decision version of Knapsack. Given a set of  $n$  items with weights  $w_1, \dots, w_n \in \mathbb{Z}^+$  and values  $v_1, \dots, v_n \in \mathbb{Z}^+$ , a capacity constraint  $C \in \mathbb{Z}^+$ , and a positive integer  $V \in \mathbb{Z}^+$ , decide if there exists a subset of items with total weight at most  $C$  and total value at least  $V$ . Prove that this decision version of Knapsack is NP-complete.

(c) Given two undirected graphs  $G$  and  $H$ , decide if  $H$  is a subgraph of  $G$ . Prove that this problem is NP-complete.

(d) 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.

(e) Given a ground set  $U = \{1, \dots, n\}$ , a collection of its subsets  $\mathcal{S} = \{S_1, \dots, S_m\}$ , and a positive integer  $k$ , the set cover problem asks if we can find a subcollection  $\mathcal{T} \subseteq \mathcal{S}$  such that  $\bigcup_{S \in \mathcal{T}} S = U$  and  $|\mathcal{T}| = k$ . Prove that set cover is NP-complete.

(f) Given a collection of integers (can be negative), decide if there is a subcollection with sum exactly 0. Prove that this problem is NP-complete.

(g) In an undirected graph  $G = (V, E)$ , each vertex can be colored either black or white. After an initial color configuration, a vertex will become black if all its neighbors are black, and the updates go on and on until no more update is possible. (Notice that once a vertex is black, it will be black forever.) Now, you are given an initial configuration where all vertices are white, and you need to change  $k$  vertices from white to black such that all vertices will eventually become black after updates. Prove that it is NP-complete to decide if this is possible.

(h) Suppose we want to allocate  $n$  items  $S = \{1, \dots, n\}$  to two agents. The two agents may have different values for each item. Let  $u_1, u_2, \dots, u_n$  be agent 1's values for those  $n$  items, and

$v_1, v_2, \dots, v_n$  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

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

and

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

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

(i) Given an undirected graph  $G = (V, E)$ , the 3-coloring problem asks if there is a way to color all the vertices by using three colors, say, red, blue and green, such that every two adjacent vertices have different colors. Prove that 3-coloring is NP-complete.

(j) Given a ground set  $U = \{1, \dots, n\}$  and a collection of its subsets  $\mathcal{S} = \{S_1, \dots, S_m\}$ , the exact cover problem asks if we can find a subcollection  $\mathcal{T} \subseteq \mathcal{S}$  such that  $\bigcup_{S \in \mathcal{T}} S = U$  and  $S_i \cap S_j = \emptyset$  for any  $S_i, S_j \in \mathcal{T}$ . Prove that exact cover is NP-complete.

**Answer 5.** It's Jan. 7 today so I guess I should only use this problem as reading materials.

**Problem 6.** 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.

**Answer 6.** I don't really understand NPC. I can't give a score. Collaborator Chatgpt, my friend.