Ceren Arkaç,   28269   , 03.01.2023,   CS301 - HW5

The problem is a variation of the 0/1 Knapsack problem.

The decision problem:
Given a set T of projects where each project is associated with a positive integer e_t denoting the amount of effort required to complete the project t and a positive integer p_t denoting the profit gained by completing the project t, is there a subset of projects such that the total amount of effort that can be spent is at most **c** and the total amount of profit gained is at least **p**?
Input:

- a set T which contains n projects
- a set E = {e_1, … , e_n } that contains the values denoting the effort for each project
- a set P = { p_1, … , p_n } that contains the values denoting the profit for each project t
- a positive integer c denoting the total amount of efforts that can be spent on some selected projects.
- a positive integer p denoting the total amount of profit that needs to be gained.

Output:

- 0 or 1. → If the answer is yes, the output is 1. Otherwise, 0.

2. To prove that it is in NP, I will show that its solvability is exponential. For a solution, we need to decide to include or not include for each project. So, try all (2^n) possible subset of projects (include or don't include for all of the n objects). Then check if the subset satisfies the conditions, i.e., total efforts of the included projects should be less than c and the total profit that is gained from the included projects is at least p. Shortly, generating all the possible solutions takes O(2^n), which requires so much time.

But, I can guess a solution and verify if the solution satisfies the conditions or not. Under the assumptions of having an infinite amount of machines that checks simultaneously if the guessed solutions are valid or not, the problem can be solved in polynomial time. For the verification, two things have to be checked. If the efforts of the projects does not exceed c and the profit gained is at least p. For a given solution, finding the e_t and p_t values of all t's that are included in the subset takes O(n). Because, E and P sets contain n elements and searching in both of them takes O(n). Whenever the e_t and p_t values are found, they are summed up to the variables like current_total_effort and current_total_profit, which takes O(1) time at each incrementation. At the end, current_total_effort and current_total_profit are compared with c and p values, respectively, which takes O(1) time. And this step gives us the output, yes or no.

As a conclusion, verification takes O(n) time, which is a polynomial time, when the size of the projects is n. Because the verification of a solution can be done in polynomial time, but the solvability of the problem takes exponential time, I can say that the problem is in NP.