Aditya Shirode, Rhythm Shah, Shrenuj Gandhi

3. [5 points each for parts (a) and (b)]

   Recall the job scheduling problem from Homework 2: Suppose n jobs, numbered $1, \ldots, n$ are to be scheduled on a single machine. Job j takes t_j time units, has an integer deadline d_j, and a profit p_j. The machine can process only one job at a time and each job must be processed without interruption (for its t_j time units). A job j that finishes before its deadline d_j receives profit p_j, while a tardy job receives no profit (and might as well not have been scheduled). Except now do not put any restriction on the number of time units for a job (except that these are positive integers). The objective (of the optimization version) is to come up with a maximum profit set of jobs that can all be scheduled by their deadlines

   (a) Formulate this problem as a decision problem and prove that the decision problem can be solved in polynomial time if and only if the optimization problem can be solved in polynomial time (using the evaluation and certificate versions as intermediaries).

   (b) Prove the the decision problem of part (a), call it **Max Profit Scheduling (MPS)**, is NP-Complete.

Answer:

(a) The decision version of the *Max Profit Scheduling Problem* is as follows -

Given n jobs, numbered $1 \ldots n$, with t_j time units, d_j integer deadlines, p_j positive profit values and a profit value k,
does there exist a set of jobs which when scheduled for the stipulated time within their deadlines yield an overall profit of atleast k?

To prove the biconditional statement : decision problem can be solved in polynomial time iff the optimization version can, we break it into two parts.

Part 1: If optimization problem can be solved in polynomial time, then the decision version can be solved in polynomial time.
From the relationship among problem types, we know that solving an optimization problem yields solution to all the others.

Thus having a polynomial time optimization problem that determines the maximum profit set of jobs, we can evaluate the maximum profit in polynomial time.
Based on this value, we can decide the decision problem's answer, thus solving it in polynomial time.

Part 2: If the decision problem can be solved in polynomial time, then the optimization version can be solved in polynomial time.

To prove polynomial time optimization problem, we need two things:
- polynomial time decision algorithm (*given*)
- polynomial time certificate algorithm

(b)