DP and Graph Algorithms
Problem Set 2 – CS 6515/4540 (Fall 2025)

Answers to problem set 2, question 6.

6 Knapsack and MDP

6.1 Knapsack

1. Let $\widetilde{v_i}$ be the modified valuations defined as $\widetilde{v_i} = K * \lfloor \frac{v_i}{K} \rfloor$. Then we know $\widetilde{v_i} \geq (v_i - K)$. Let the optimal set of items chosen by the PTAS is $\widetilde{R}$ and the optimal pseudo-poly time algorithm gives us R. It is not necessary that R and $\widetilde{R}$ be the same, but we can say for sure that $\widetilde{v_i} \leq v_j$ for some matching (i, j) pairs in $\widetilde{R}$ and R. If this were not the case then it would be possible to choose some other element $j = i$ in R such that $\widetilde{v_i} \leq v_j$ and since both sets have to satisfy the same budget constraints we have found a better optimal solution, which would be a contradiction.

   Moving forward with this result we get, $V' = \sum_{i \in R} v_i$ and $V = \sum_{i \in \widetilde{R}} \widetilde{v_i}$.

   Since, $\widetilde{v_i} \leq v_j$ for some matching (i, j) pairs in $\widetilde{R}$ and R, we get

$$V \leq V' => \frac{V}{V'} \leq 1$$

   We also know,

$$\sum_{i \in \widetilde{R}} \widetilde{v_i} \geq \sum_{i \in R} (v_i - K)$$

$$V \geq V' - nK$$

$$\frac{V}{V'} \geq (1 - \frac{nK}{V'})$$

$$\frac{V}{V'} \geq (1 - \frac{nK}{\max_{i \in R} v_i})$$

   More generally,

$$\frac{V}{V'} \geq (1 - \frac{nK}{\max v_i})$$

2. With the above result, we can choose

$$(1 - \frac{nK}{\max v_i}) \geq 1 - \epsilon$$

$$\frac{nK}{\max v_i} \leq \epsilon$$

$$K \leq \frac{\epsilon * \max v_i}{n}$$

Let's choose $K = \frac{\epsilon * \max v_i}{n}$. The time complexity of the pseudo-polytime algorithm is $O(n^2 \max v_i)$, in our case we run the PTAS for the modified values $\lfloor \frac{v_i}{K} \rfloor$, this gives us the time complexity,

$$O(n^2 \max \lfloor \frac{v_i}{K} \rfloor) => O(n^2 \frac{\max v_i}{K}) => O(\frac{n^3}{\epsilon})$$

Thus, with our constraint of a constant ϵ the above choice of K gives us a polynomial run time.

3. We have,

$$\frac{V}{V'} \geq (1 - \epsilon)$$

Now with the constraint on $\epsilon < \frac{1}{n \max v_i}$, we get

$$\frac{V}{V'} \geq (1 - \frac{1}{n \max v_i})$$

but the largest possible difference between V and V' is nK, so this implies $nK < 1$. Therefore, the approximation algorithm is computing the exact optimal result and we have managed to find a polynomial time solution for the original optimization Knapsack problem. If we can find the solution to the optimization variant in poly time then we can also find a solution for the decision variant in poly time. The decision variant of the Knapsack problem is considered an NP-Complete problem, by finding a polynomial time algorithm for an NP-Complete problem we have found polynomial time algorithms for all NP problems. Hence, we have proved $P = NP$!

6.2 MDP

We can solve this using the DP based algorithm discussed in class. Let's first define the base case.

- When $T = 0$ or 0 pulls left, irrespective of state we're in, the reward is always 0. This is because we can't take any actions now.
- When $T = 1$ or 1 pull left and given that any action from state S_1 or S_2 gives 0 reward. We define:
  - $R_{a_i}(S_k) = 0 \quad \text{for k} \in \{1, 2\}$
  - $R_{a_i}(S_3) = 10$
  - $R_{a_i}(S_4) = 100$

  With the above base case, to solve fo $T = 2$ we observe that it's only possible to reach S_4 in 2 steps, therefore we can never receive the reward $R_{a_i}(S_4)$. The other state that gives us some reward is S_3, but we get this reward only if we reach S_3 with at least 1 time step remaining. So, the optimal policy is to pull action a_2 at starting state S_1 and reach S_3 (an absorbing state), then we can take any action a_i to receive a reward of $R_{a_i}(S_3) = 10$.