Exercise 3

Lecturer: Shahar Dobzinski *Scribe: Anton Bar*

1 Continuation of Knapsack

There is a group of n items $\{a_1, a_2, ..., a_n\}$. Each item has size and profit. The goal is to find the most profitable set of items with overall weight $\leq B$.

We saw a dynamic programming algorithm that solves the problem in time $O(n^2P)$, where P is the profit of the solution. The problem is that the representation of number of profit P takes $log(P)$ bits so the representation of the entire problem will take $O(n^2 logP)$. We have a pseudo-polynomial algorithm. It works at cost $O(n^2P)$, where the complexity is polynomial only in terms of value of P, and not its representation as we would like to have. Given an unbounded profit, our pseudo-polynomial algorithm is, in fact, exponential.

In order to deal with the representation problem we can sacrifice accuracy and gain speed. For example we can divide all profit values by 1 million (e.g. $100, 800, 578 \rightarrow 100$).

1.1 Approximated Knapsack Algorithm

Lets define:

1) $K = \frac{\epsilon P}{n}$

2) $\forall i : profit'(a_i) = \left\lfloor \frac{profit(a_i)}{K} \right\rfloor$

3) $P = max_i(profit'(a_i))$

The algorithm finds maximal set using the Dynamic programming algorithm (ALG).

Lemma 1 $|ALG| \geq (1-\epsilon)|OPT|$ - *the algorithm is* $(1-\epsilon)$*-approximation.*

Proof of Lemma 1: OPT contains at most n items. While diving by K and rounding we lose at most K, thus we lose at most $n \cdot K$.

$|OPT| \geq Kprofit'(ALG) \geq Kprofit'(OPT) \geq profit(OPT) - nK = profit(OPT) - n\frac{\epsilon P}{n} \geq (1-\epsilon)profit(OPT)$. ■

Claim 2 *The running time of the algorithm is polynomial in* $\frac{1}{\epsilon}$ *and in* n.

Proof of Claim 1: $P_{new} = \left\lfloor \frac{P}{K} \right\rfloor = \left\lfloor \frac{P}{\frac{\epsilon P}{n}} \right\rfloor = \left\lfloor \frac{n}{\epsilon} \right\rfloor \Rightarrow O(n^2 P_{new}) = O(\frac{n_3}{\epsilon})$. ■

2 Set Cover - Randomized Rounding

2.1 Definition of LP

Minimize: $\sum x_s c(s)$. **Subject to:**

- $\forall e \in U : \sum_{s|e \in s} x_s \geq 1$
- $\forall s : x_s \geq 0$

2.2 Reminders

Markov's Inequality:
If x is positive random variable, then $\forall t \geq 0 : Pr[x \geq t] \leq \frac{E[x]}{t}$.

Intuition:

For example: if $E[x] = 1$ and $t = 100$, and provided that the distribution is uniform, the inequality says that the chance that $x \geq 100$ is at most $\frac{1}{100}$. Assume the negation, so if, for instance, $x_1 = 101, x_2 = ... = x_{100} = 0$ then $E[x] > 1$ in contradiction to the given value.

Proof:

Define: $y = \begin{cases} d, & x > d \\ 0, & otherwise \end{cases}$

Then $0 \leq y \leq x$. Additionally, it follows immediately from the definition that Y is a random variable. Computing the expected value of Y, we have that: $E[x] \geq E[y] = d \cdot Pr[x > d]$. ■

Union Bound: Let $B_1, ..., B_n$ be events happening with probabilities $p_1, ..., p_n$. Union bound defines the probability of the case when no B_i has happened: $Pr[\overline{B_1} \wedge ... \wedge \overline{B_n}] \leq 1 - \sum_{i=1}^{n} p_i$.

2.3 Randomized SC Algorithm

For $i = 1, ..., klogn$, where $kgeq0$ is constant, we add s to c_i with probability x_s.
Return $ALG = \bigcup c_i$.

The expectation is: $E[\bigcup c_i] = k \cdot logn$.

Analysis:

Lemma 3 *Provided B_1, event when the inequality $\sum_{s \in ALG} c(s) \geq 4k \cdot logn \cdot OPT_F$ is true, then probability of this event $Pr(B_1) \leq \frac{1}{4}$.*

Lemma 4 *Provided B_2, event when $\exists e \in \cup$, so that, $e \notin ALG$, then probability of this event $Pr(B_2) < \frac{1}{4}$.*

Claim 5 *From Lemma 3 and Lemma 4 follows that the algorithm is 2-approximation.*

Proof of Claim 5: By the union bound we know that $Pr[\overline{B_1} \wedge \overline{B_2}] \leq 1 - \frac{1}{4} \cdot 2 = \frac{1}{2}$. ■

Proof of Lemma 3: Lets set some fixed i. Expectation of $c(c_i)$, cost of c_i, is: $E[c(c_i)] = \sum_{s \in S} Pr[s \in c_i]c(s) = \sum_{s \in S} x_s c(s) = OPT_F$.

$E[ALG] \leq \sum_{i=1}^{klogn} E[c(c_i) = k \cdot logn \cdot OPT_F$.

Thus, by Markov's inequality, $Pr[E(ALG)] \geq 4klogn \leq \frac{1}{4}$. ■

Proof of Lemma 4: Lets set some fixed i and will find the probability of an uncovered edge: $e \notin c_i$. Without loss of generality we will examine sets covering $e : s_1, ..., s_r$.

$Pr[e \notin c_i] = \prod_{t=1}^{r} 1 - x_{s_t} \leq (1 - \frac{1}{r})^r \leq \frac{1}{e}$. Note: the last two inequalities are true because of the condition of LP: $\sum s_t \geq 1$ and some math analysis.

Probability of e not covered by ALG: $Pr[e \notin ALG] \leq \frac{1}{e} \cdot \frac{1}{e} \cdot ... \cdot \frac{1}{e}$. Note: $\frac{1}{e}$ is repeated $klogn$ times, one for each c_i. For k large enough $(\frac{1}{e})^{klogn} \leq \frac{1}{4n}$.

Note that for each specific item e: $Pr[e \notin ALG] \leq \frac{1}{4n}$.

Denote B_e as event of $e \notin ALG$ for all e, and $B_t = min_{e \in \cup} B_e$: $Pr[\exists e, e \notin ALG] = 1 - Pr[\overline{B_1} \wedge ... \wedge \overline{B_{|\cup|}}] \leq \sum_{t=1}^{|\cup|} Pr[B_t] \leq n \cdot \frac{1}{4n} = \frac{1}{4}$, provided k large enough. ■

Note: this approximation is optimal. It's known that there is no better approximation.