EECS 336: Lecture 17: Introduction to Algorithms

Approximation Algorithms approximation, metric TSP, knapsack

Reading: 11.8

Last Time:

- approximation
- metric TSP
- knapsack

Today:

- pseudo polynomial time
- knapsack $(1+\epsilon)$ approx.

Def: $\mathcal{A}$ is a β-approximation if the value of its solution is at least OPT$/\beta$ (maximization problems)

Recall: knapsack problem

input:

- n objects
- size s_i (non-negative real number)
- values v_i
- capacity C.

output: subset S that

- fits: $\sum_{i \in S} s_i \leq C$
- maximizes values: $\sum_{i \in S} v_i$.

Recall: Integer Knapsack Dynamic Program

"s_is and C are integers"

Step I: identify subproblem in English

$\text{OPT}(i, D)$

$=$ "value of optimal size D knapsack on $\{i, ..., n\}$"

Step II: write recurrence

$\text{OPT}(i, D)$

$= \max(\underbrace{v_i + \text{OPT}(i+1, D - s_j)}_{\text{if } s_i \leq D}, \text{OPT}(i+1, D)$

Step VI: runtime

$$\begin{aligned} T(n, C) &= O(\# \text{ of subprobs} \times \text{cost per subprob}) \\ &= O(nC). \end{aligned}$$

Pseudo-polynomial Time

"polynomial if numbers in input are written in unary (not binary)"

Thm: Integer Knapsack DP is pseudo polynomial time.

Polynomial Time Approximation Scheme (PTAS)

"for any constant ϵ, get $(1+\epsilon)$-approximation algorithm in polynomial time."

Note: often pseudo-polynomial time alg can be converted into PTAS by rounding.

Knapsack PTAS

Goal: output $(1+\epsilon)$-approximation to optimal knapsack value.

Idea: round so that numbers are integers in range from 0 to poly(n).

Recall: for old knapsack dynamic program, need sizes to be integer, but approximation would allow for rounding values not sizes.

Approach:

1. write new dynamic program that is pseudo-polynomial in values not capacity. $O(n^2 v_{\max})$
2. divide values by $\epsilon v_{\max}/n$ and round up. (range from 0 to n/ϵ.)
3. solve dynamic program on rounded values.

Value-based Knapsack DP

Idea: instead of maximizing values, let's minimize size.

Part I: Subproblem

$\text{MinSize}(i, V)$ = smallest total size of subset of $\{i, ..., n\}$ with total value at least V.

Part II: Recurrence

$\text{MinSize}(i, V) = \max\{s_i + \text{MinSize}(i+1, \max\{V - v_i, 0\}), \text{MinSize}(i+1, V)\}$

Part III: Invocation

1. $V \leftarrow \sum_i v_i$
2. while $\text{MinSize}(1, V) >$ C

   $V \leftarrow V - 1$
3. output V.

Part IV: Base case

$$\text{MinSize}(n+1, V) = \begin{cases} 0 & \text{if } V = 0 \\ \infty & \text{o.w.} \end{cases}$$

Theorem: ALG has pseudo-polynomial runtime $O(n^2 v_{\max})$ if v_is are integer,

Proof: table size $= n \times \sum_i v_i \leq n \times n v_{\max}$

Polynomial Time Approximation Scheme

Algorithm: Knapsack $(1+\epsilon)$-approx

1. round v_i up to multiple of $\epsilon v_{\max}/n \to \tilde{v}_i$
2. divide $\tilde{v}_i$ by $\epsilon v_{\max}/n \to \hat{v}_i$ (integer)
3. solve integral knapsack on $\hat{v}_1, ..., \hat{v}_n \to S$
4. output $\max(v_{\max}, \sum_{i \in S} v_i)$

Correctness

Lemma: ALG is optimal for $\hat{v}_i$s and $\tilde{v}_i$s.

Proof: via correctness of DP.

Lemma: ALG is polynomial in n and $1/\epsilon$

Proof:

- $\hat{v}_{\max} = v_{\max} \times \frac{n}{\epsilon v_{\max}} = n/\epsilon$
- runtime is $O(n^2 v_{\max}) = O(n^3/\epsilon)$.

Lemma: ALG is $(1+\epsilon)$-approx for v_is.

Proof:

1. lower bound on OPT

$$\begin{aligned} OPT &= \sum_{i \in S^*} v_i && \text{(OPT's actual values)} \\ &\le \sum_{i \in S^*} \tilde{v}_i && \text{(OPT's rounded values)} \\ &\le \sum_{i \in S} \tilde{v}_i && \text{(ALG's rounded values)} \end{aligned}$$

   Last step by optimality of ALG on $\tilde{v}$s and $\hat{v}$s.

2. upper bound on algorithm

   - bound 1:

$$\begin{aligned} ALG &\ge \sum_{i \in S} v_i && \text{(ALG's actual values)} \\ &= \sum_{i \in S} \tilde{v}_i - \sum_{i \in S} \underbrace{(\tilde{v}_i - v_i)}_{\le \epsilon v_{\max}/n} \\ &\ge \sum_{i \in S} \tilde{v}_i - n \times \epsilon v_{\max}/n \\ &= \sum_{i \in S} \tilde{v}_i - \epsilon v_{\max} \end{aligned}$$

   - bound 2: ALG $\ge v_{\max}$.

3. combine:

$$\begin{aligned} \text{ALG} &\ge \underbrace{\sum_{i \in S} \tilde{v}_i}_{\ge \text{OPT}} - \epsilon \underbrace{v_{\max}}_{\le \text{ALG}} \\ &\ge \text{OPT} - \epsilon \text{ALG} \end{aligned}$$

So $(1+\epsilon)\text{ALG} \ge \text{OPT}$.

QED

Complexity of Approximation

Def: APX = class of problems with constants approximations

Def: PTAS = class of problems with PTASs.

DRAW PICTURE of

$$P \le PTAS \le APX \le NP$$