10 Approx Knapsack, DP on trees/DAGs

Tuesday, February 11, 2025 13:52

$O(n B)$ — #items, size of bag

1) $O(nV)$ — sum of values $V = \sum_{i=1}^{n} v[i]$

2) $1+\varepsilon$ approximation $O(n^2/\varepsilon)$

Input: $S[1...n]$ $S[i]$ = size of ith item
$v[1...n]$ $v[i]$ = value of ith item
B size of bag

Find $S \subseteq \{1...n\}$ where $\sum_{i \in S} s[i] \leq B$

maximize $\sum_{i \in S} v[i]$

$O(nV)$ $V := \sum_{i=1}^{n} v[i]$ fast if values are small

vs

$O(nB)$ fast if bag is small

last time

$T[i, b]$ = max value if we have first i items available as option
b is bag size

today

$T[i, v]$ = min bag size needed to pack v amount of value
if we have first i items available as option to pack.

base case

$T[i, 0] = 0$ // no value -> no bag

$T[0, w] = \infty$ $\forall w$ // no items available, so store worst possible size of bag

recursion

$T[i, w] = \min\left(\underline{T[i-1, w]}, \underline{T[i-1, w - v[i]] + s[i]}\right)$

if we do not pack i-th item — if we do pack i-th item if we do not pack i-th item   if we do pack i-th item

result:

return largest $W \le \sum_{i=1}^{n} v[i]$ such that $T[n, W] \le B$

```
LW = 0
for w = 0 ... Σ v[i]
    if T[n,w] ≤ B
        LW = w
return LW
```

Complexity $O(n \cdot \sum v[i])$ to fill the table

then $O(\sum_{i=1}^{n} v[i])$ to find value to return

$\Rightarrow O(n \sum_{i=1}^{n} v[i])$

Next $(1-\varepsilon)$-approximation in $O(n^2/\varepsilon)$ time.

OPT := maximum value that can be packed in a bag of size B

$\tilde{W}$:= value returned by algorithm

We want $OPT \ge \tilde{W} \ge (1-\varepsilon) \cdot OPT$ eg for $\varepsilon = 0.01$ we get a solution that is at most 1% off

Idea: make values $v[1] \ldots v[n]$ smaller

Input: $S[1 \ldots n]$
$v[1 \ldots n]$
B, ε

$v_{max} = \max_{1 \le i \le n} v[i]$

$\tilde{v}[i] = \left\lfloor \frac{v[i]}{k} \right\rfloor$ where $k = \frac{v_{max}}{n} \cdot \varepsilon$

$$O(n \cdot \sum \tilde{v}[i]) = O\left(n \sum_i \left(\frac{v[i]}{v_{max}} \cdot n/\varepsilon\right)\right) \le O(n^2/\varepsilon)$$

return PrevAlgo$(S[1..n], \tilde{v}[1...n], B) \cdot k$ // $\tilde{W}$

$\tilde{W} \le OPT$ because of division and multiplication by k and rounding down

Let $\tilde{S} \subseteq \{1 \ldots n\}$ is what was used by PrevAlgo to get $\sum_{i \in \tilde{S}} \tilde{v}[i] \cdot k = \tilde{W}$

Let $S \subseteq \{1 \ldots n\}$ be the optimal solution: $\sum_{i \in S} v[i] = OPT$

Let $S \subseteq \{1 \dots n\}$ be the optimal solution: $\sum_{i \in S} v[i] = OPT$

Claim: $\sum_{i \in \tilde{S}} v[i] \geq (1-\varepsilon) \cdot \sum_{i \in S} v[i]$

$\sum_{i \in \tilde{S}} \tilde{v}[i] \cdot k \geq (1-\varepsilon) \cdot \sum_{i \in S} v[i]$

$$
\begin{aligned}
\sum_{i \in \tilde{S}} v[i] \geq \sum_{i \in \tilde{S}} \tilde{v}[i] \cdot k \geq k \cdot \sum_{i \in \tilde{S}} \tilde{v}[i] &\geq k \cdot \sum_{i \in S} \tilde{v}[i] \\
&= k \cdot \sum_{i \in S} \left\lfloor \frac{v[i]}{k} \right\rfloor \\
&\geq k \cdot \sum_{i \in S} \left(\frac{v[i]}{k} - 1 \right) \\
&= \left(\sum_{i \in S} v[i] \right) - \sum_{i \in S} k \\
&= OPT - |S| \cdot k \\
&= OPT - |S| \cdot \frac{V_{max}}{n} \cdot \varepsilon \\
&\geq OPT - V_{max} \cdot \varepsilon \\
&\geq OPT - OPT \cdot \varepsilon \\
&= (1-\varepsilon) \cdot OPT
\end{aligned}
$$

$\geq OPT - \underline{n} \cdot \frac{V_{max}}{\underline{n}} \cdot \varepsilon$

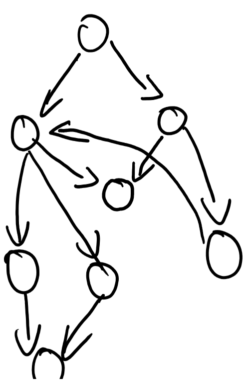
$V_{max} \leq OPT$

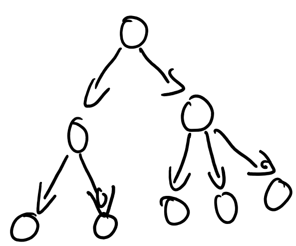
by assumption $S[i] \leq B \; \forall i$ otherwise there is no point in having that item as option.

$\Rightarrow OPT \geq v[i] \quad \forall i$

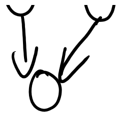
$OPT \geq V_{max}$

DAG directed acyclic graph





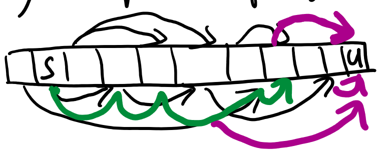
For Dyn. Prog. we often cut off first or last piece of input to argue some



first or last piece of input to argue some recursion.

For DAGs

1) compute topological order



$D[v]$ = distance from s to v

$$D[u] = \min_{(v,u)\in E} \left(D[v] + c_{vu} \right)$$

$\Rightarrow O(|E|)$ time to fill table

for u in topological order

$$D[u] = \min_{(v,u)\in E} \left(D[v] + c_{vu} \right)$$