FPTAS on 0/1 Knapsack 2

0/1 Knapsack

- n objects
- p_k profit for object k
- w_k weight of object k
- b capacity

Maximize

$$z = \sum_{k=1}^{n} p_k x_k$$

subject to:

$$\sum_{k=1}^{n} w_k x_k \leq b$$

$$x_k \in 0, 1 \;\; k = 0, ..., n$$

0/1 Knapsack with DP

- s = remaining capacity
- solution is $f_1(b)$

Recursion

$$f_k(s) = max \begin{cases} p_k + f_{k+1}(s - w_k) & x_k = 1 \\ f_{k+1}(s) & x_k = 0 \end{cases}$$

Base

$$f_n(s) = \begin{cases} 0 & s < w_n \\ p_n & s \geq w_n \end{cases}$$

Time complexity: $O(nb)$

0/1 Knapsack with Volume constraint

- n objects
- p_k profit for object k
- w_k weight of object k
- v_k volume of object k
- b tot. capacity
- V tot. volume

Maximize

$$z = \sum_{k=1}^{n} p_k x_k$$

subject to:

$$\sum_{k=1}^{n} w_k x_k \leq b$$

$$\sum_{k=1}^{n} v_k x_k \leq V$$

$$x_k \in 0, 1 \;\; k = 0, ..., n$$

0/1 Knapsack with Volume constraint - DP

- s = remaining capacity, remaining volume
- solution is $f_1(\langle b, V \rangle)$

Recursion

$$f_k(s) = max \begin{cases} p_k + f_{k+1}(s - \langle w_k, v_k \rangle) & x_k = 1 \\ f_{k+1}(s) & x_k = 0 \end{cases}$$

Base

$$f_n(s) = \begin{cases} 0 & s < \langle w_n, v_n \rangle \\ p_n & s \geq \langle w_n, v_n \rangle \end{cases}$$

Time complexity: $O(nbV)$

0/1 Knapsack - DP version 2

- recursion on profit
- s = remaining units of profit (0...P)
- idea: find a set of objects with maximum total profit ($\leq s$) and $\langle$ weight, volume $\rangle \leq \langle b, V \rangle$

Recursion

$$f_k(s) = min \begin{cases} \langle w_k, v_k \rangle + f_{k+1}(s - p_k) & x_k = 1 \\ f_{k+1}(s) & x_k = 0 \end{cases}$$

Base

$$f_n(s) = \begin{cases} \langle +\infty, +\infty \rangle & s \neq p_n \\ \langle w_n, v_n \rangle & s = p_n \\ \langle 0, 0 \rangle & s = 0 \end{cases}$$

Time complexity

- Time complexity is **pseudo-polynomial**: $O(nP)$
- If all objects fit $\rightarrow$ $\mathrm{P} = \sum_{i=0}^{n} p_i$

Time complexity

- Time complexity is **pseudo-polynomial**: $O(nP)$
- If all objects fit $\rightarrow$ P $= \sum_{i=0}^{n} p_i$
- If P is small $\rightarrow$ *polynomial* time

FPTAS

- Fully-Polinomial time approximation scheme
- Scale profits down so that time is polynomial
- Solve scaled instance of the problem