

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:

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

$$x_k \in \{0, 1\} \quad k = 1, \dots, n$$

## 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:

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

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

$$x_k \in \{0, 1\} \quad k = 1, \dots, n$$

## 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 ( $0 \dots P$ )
- ▶ idea: find a set of objects with maximum total profit ( $\leq s$ ) and  $\langle \text{weight, volume} \rangle \leq \langle b, V \rangle$

### 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**:  $O(nP)$
- ▶ Theoretical upper bound of  $P \rightarrow np_{max}$

# Time complexity

- ▶ Time complexity is **pseudo-polynomial**:  $O(nP)$
- ▶ Theoretical upper bound of  $P \rightarrow np_{max}$
- ▶ If  $P$  is small  $\rightarrow$  *polynomial* time



# FPTAS - I

- ▶ Fully-Polynomial time approximation scheme
- ▶ Scale profits down so that time is polynomial
- ▶ Solve scaled instance  $I$  of the problem
- ▶ Solution  $A(I)$  is a  $\varepsilon$ -approximation of the optimal solution

$$A(I) \geq (1 - \varepsilon)OPT(I)$$

## FPTAS - II

given  $\varepsilon > 0$

1. let  $p_{max} = \max\{p_0, \dots, p_n\}$
2. let  $K = \frac{\varepsilon p_{max}}{n}$
3. for each object  $i$ , define  $p'_i = \frac{p_i}{K}$
4. Solve scaled instance with dynamic programming
5. Return solution  $S'$

Time Complexity:

$$O(nP') = O(n * np'_{max}) = O(n^2 * \frac{p_{max}}{K}) = O(n^3 * \frac{1}{\varepsilon})$$

## FPTAS - III

$$S' \geq (1 - \varepsilon)OPT(I)$$

### Proof

Stocazz [<http://math.mit.edu/~goemans/18434S06/knapsack-katherine.pdf>]