FPTAS - 0/1 Knapsack with Volume constraint

Barbano, Conforti, Gerbaldo

0/1 Knapsack - Simple version

- 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:

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

$$\sum_{k=1}^{n} v_k x_k \leq V, \quad x_k \in 0, 1 \ k = 0, ..., n \tag{2}$$

0/1 Knapsack with Volume constraint - DP

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

Recursion

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

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 alternative version

- 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

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

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-Polinomial time approximation scheme
- Scale profits down so that time is polynomial (scale factor ε)
- Solve scaled instance I of the problem
- Solution A(I) is a ε-approximation of the optimal solution

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

FPTAS - II

given $\varepsilon > 0$

1. let $p_{max} = max\{p_0, ..., p_n\}$
2. let $K = \frac{\varepsilon p_{max}}{n}$
3. for each object i, define $p'_i = \lfloor \frac{p_i}{K} \rfloor$
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

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

Proof 1/2

- For every object $i \rightarrow Kp'_i \leq p_i$
- $p_i - Kp'_i = K$ **at most**
- Be O the optimal set with the maximum profit $\rightarrow$ the maximum difference between the profit $P(O)$ of the original instance and the profit $P'(O)$ of the scaled instance is:

$$(a.) \qquad P(O) - KP'(O) \leq nK$$

- Be S' the solution of the scaled instance found by dynamic programming $\rightarrow P(S')$ at least as good as $KP'(O)$

FPTAS - III

Proof 2/2

$$
\begin{aligned}
P(S') &\geq KP'(O) && (3)\\
&\geq P(O) - nK \quad (\mathit{for\ a.}) && (4)\\
&= OPT - \varepsilon p_{max} && (5)
\end{aligned}
$$

Since $OPT \geq p_{max}$

$$OPT - \varepsilon p_{max} \geq OPT - \varepsilon OPT = (1-\varepsilon)OPT$$

And so

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