FPTAS for the 0–1 Knapsack Problem

Cheng Xin

November 22, 2025

Definition (FPTAS). An algorithm is a Fully Polynomial-Time Approximation Scheme (FPTAS) for a maximization problem if, for every $\varepsilon \in (0,1)$, it outputs a solution S satisfying

$$\text{value}(S) \geq (1-\varepsilon)\,\text{OPT},$$

and its running time is polynomial in both n and $1/\varepsilon$.

The 0–1 Knapsack problem is NP-hard, but it admits an FPTAS.

0-1 Knapsack Problem: We are given $[n]$ items with values $v : [n] \to \mathbb{R}_{>0}$, weights $w : [n] \to \mathbb{R}_{>0}$, and a knapsack capacity W.

The value-based DP algorithm maintains

$$DP[i][v] = \text{minimum weight needed to obtain total value exactly } v \text{ using items } 1,\ldots,i.$$

with recurrence

$$DP[i][v] = \min\big(DP[i-1][v],\ w_i + DP[i-1][v-v_i]\big).$$

We adopt the convention that $DP[i][v] = +\infty$ whenever $v < 0$. The algorithm returns the largest v such that $DP[n][v] \leq W$. Its running time is $\Theta(n\sum_i v_i)$—pseudo-polynomial because it depends on the magnitude of values v_i's.

Idea of the FPTAS. We compress the value domain by rounding each v_i down to a grid multiple of width $\delta > 0$. The smaller δ is, the more accurate the approximation but the larger the DP table. Later we choose δ so that the rounding loss is at most $\varepsilon \cdot \text{OPT}$ while the DP algorithms remains polynomial time complexity.

DP on the discretized grid.
Set the discretized value grid as:

$$\mathcal{V} = \{0, \delta, 2\delta, \ldots, \hat{v}_{total}\}, \qquad \hat{v}_{total} = \left\lfloor \frac{\sum_i v_i}{\delta} \right\rfloor \delta.$$

Rounding (with undetermined width parameter δ for now)
Define the rounded values $\hat{v}_i$ as:

$$\hat{v}_i = \delta \left\lfloor \frac{v_i}{\delta} \right\rfloor \in \mathcal{V},$$

Observe that

$$v_i - \delta < \hat{v}_i \leq v_i.$$

Initialize

$$DP[0][0] = 0, \qquad DP[0][v] = +\infty \text{ for } v \neq 0.$$

For $i = 1,\ldots,n$ and $v \in \mathcal{V}$, apply the recurrence:

$$DP[i][v] = \min\big(DP[i-1][v],\ w_i + DP[i-1][v-\hat{v}_i]\big).$$

Let $\hat{S}$ be the set achieving the largest $v \in \mathcal{V}$ with $DP[n][v] \leq W$, and define

$$ALG = \sum_{i \in \hat{S}} v_i.$$

Approximation guarantee. Let S^* be optimal:

$$\mathrm{OPT} = \sum_{i \in S^*} v_i.$$

Since $\hat{v}_i > v_i - \delta$,

$$\sum_{i \in S^*} \hat{v}_i > \mathrm{OPT} - |S^*|\delta \geq \mathrm{OPT} - n\delta.$$

The rounded DP optimizes $\hat{v}$, so

$$\sum_{i \in \hat{S}} \hat{v}_i \geq \sum_{i \in S^*} \hat{v}_i.$$

Since $\hat{v}_i \leq v_i$,

$$ALG \geq \sum_{i \in \hat{S}} \hat{v}_i \geq \mathrm{OPT} - n\delta.$$

Choosing δ. To ensure $ALG \geq (1-\varepsilon)\mathrm{OPT}$, it suffices that

$$n\delta \leq \varepsilon\, \mathrm{OPT}.$$

Since $\mathrm{OPT} \geq v_{\max}$ (assuming $\forall i, w_i \leq W$, otherwise just ignore the over-weighted items),

$$n\delta \leq \varepsilon v_{\max} \quad \Longrightarrow \quad \delta = \frac{\varepsilon v_{\max}}{n}.$$

Thus

$$ALG \geq (1-\varepsilon)\mathrm{OPT}.$$

Running time.

$$|\mathcal{V}| = O(\hat{v}_{total}/\delta) = O(\left\lfloor \frac{\sum_i v_i}{\delta} \right\rfloor) = O\left(\frac{n^2}{\varepsilon}\right), \qquad T = O(n|\mathcal{V}|) = O\left(\frac{n^3}{\varepsilon}\right).$$

This proves the algorithm is an FPTAS.

Pseudocode for the FPTAS

Algorithm 1 FPTAS for 0–1 Knapsack (value rounding)

Compute $v_{\max} = \max_i v_i$ and set $\delta = \frac{\varepsilon v_{\max}}{n}$.
for $i = 1$ to n **do**
  $\hat{v}_i \leftarrow \delta \left\lfloor \frac{v_i}{\delta} \right\rfloor$
end for
Build value grid $\mathcal{V} = \{0, \delta, 2\delta, \ldots, \hat{v}_{total}\}$ up to $\hat{v}_{total} = \left\lfloor \frac{\sum_i v_i}{\delta} \right\rfloor \delta$.
Initialize $DP[0][0] = 0$ and $DP[0][v] = +\infty$ for $v > 0$.
for $i = 1$ to n **do**
  for each $v \in \mathcal{V}$ **do**
    $DP[i][v] = \min\big(DP[i-1][v],\ w_i + DP[i-1][v - \hat{v}_i]\big)$
  end for
end for
Return $\hat{S}$ = largest feasible value with total weights $\leq W$.