

LEC016 Knapsack Problem

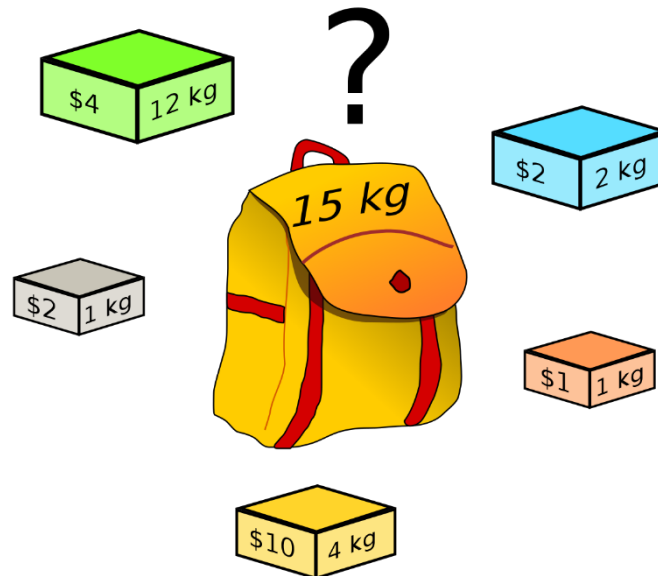
VG441 SS2021

Cong Shi
Industrial & Operations Engineering
University of Michigan

Knapsack

- n items $\{1, \dots, n\}$
- Each item i has a value v_i and size s_i
- A capacity B (for sizes)

Objective: We wish to pick a maximum-value subset S from n items (without exceeding the capacity constraint).



MILP Formulation

- n items $\{1, \dots, n\}$
- Each item i has a value v_i and size s_i
- A capacity B (for sizes)
- Find a max-value subset S from n items

Let x_i be a binary decision variable of whether to include the item i

$$\begin{array}{ll}\mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & x_i \in \{0, 1\}, \forall i\end{array}$$

LP Relaxation (Fractional Knapsack)

- n items $\{1, \dots, n\}$
- Each item i has a value v_i and size s_i
- A capacity B (for sizes)
- Find a max-value subset S from n items

Let x_i be a binary decision variable of whether to include the item i

$$\begin{array}{ll} \mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & x_i \in \{0, 1\}, \forall i \end{array} \quad \longrightarrow \quad \begin{array}{ll} \mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & 0 \leq x_i \leq 1, \forall i \end{array}$$

LP Relaxation: allowing for fractional allocation

LP Relaxation (Fractional Knapsack)

- n items $\{1, \dots, n\}$
- Each item i has a value v_i and size s_i
- A capacity B (for sizes)
- Find a max-value subset S from n items

Let x_i be a fraction of item i to be included

$$\begin{array}{ll}\mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & 0 \leq x_i \leq 1, \forall i\end{array}$$

This is easy: greedy would suffice!

Re-index $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \geq \dots \geq \frac{v_n}{s_n}$

The solution would be $(1, 1, \dots, 1, \alpha, 0, \dots, 0, 0)$
with $0 \leq \alpha < 1$

Back to (Integer Knapsack)

- n items $\{1, \dots, n\}$
- Each item i has a value v_i and size s_i
- A capacity B (for sizes) Assume $B \geq s_i$ for each i
- Find a max-value subset S from n items

$$\begin{array}{ll}\mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & x_i \in \{0, 1\}, \forall i\end{array}$$

Re-index $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \geq \dots \geq \frac{v_n}{s_n}$

The solution would be $(1, 1, \dots, 1, \alpha, 0, \dots, 0, 0)$

Trim

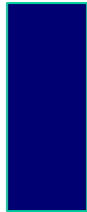
Greedy solution is $(1, 1, \dots, 1, 0, 0, \dots, 0, 0)$

However, this can be arbitrarily bad!

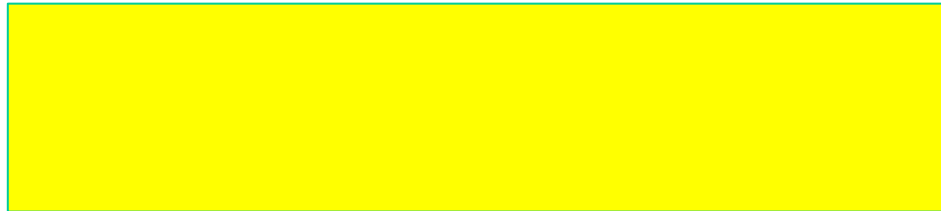
A Bad Example

- Greedy can be arbitrarily bad for 0-1 Knapsack

Capacity $B = 10000$



Item 1: $s_1 = 1, v_1 = 100$



Item 2: $s_2 = 10000, v_2 = 10000$

A 2-Approximation Algorithm

- n items $\{1, \dots, n\}$
- Each item i has a value v_i and size s_i
- A capacity B (for sizes) Assume $B \geq s_i$ for each i
- Find a max-value subset S from n items

$$\begin{array}{ll}\mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & x_i \in \{0, 1\}, \forall i\end{array}$$

Re-index $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \geq \dots \geq \frac{v_n}{s_n}$

The solution would be $(1, 1, \dots, 1, \alpha, 0, \dots, 0, 0)$

Choose the better of the two:

SOL1 = $(1, 1, \dots, 1, 0, 0, \dots, 0, 0)$

SOL2 = $(0, 0, \dots, 0, 1, 0, \dots, 0, 0)$

A 2-Approximation Algorithm

$$\begin{array}{ll}\max & \sum_{i=1}^n v_i x_i \\ \text{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & x_i \in \{0, 1\}, \forall i\end{array}$$

$$\begin{array}{ll}\max & \sum_{i=1}^n v_i x_i \\ \text{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & 0 \leq x_i \leq 1, \forall i\end{array}$$

Re-index $\frac{v_1}{s_1} \geq \frac{v_2}{s_2} \geq \dots \geq \frac{v_n}{s_n}$

The greedy solution would be $(1, 1, \dots, 1, \alpha, 0, \dots, 0, 0)$

$$\text{So } OPT(MILP) \leq OPT(LP) \leq v_1 + v_2 + \dots + v_{k-1} + v_k$$

Choose the better of the two:

$$\text{SOL1} = (1, 1, \dots, 1, 0, 0, \dots, 0, 0)$$

$$\text{SOL2} = (0, 0, \dots, 0, 1, 0, \dots, 0, 0)$$

So we are getting

$$\max(v_1 + v_2 + \dots + v_{k-1}, v_k) \geq 0.5 OPT(LP) \geq 0.5 OPT(MILP)$$

Is there a better solution?

$$\begin{array}{ll}\mathbf{max} & \sum_{i=1}^n v_i x_i \\ \mathbf{s.t.} & \sum_{i=1}^n s_i x_i \leq B \\ & x_i \in \{0, 1\}, \forall i\end{array}$$

Our goal (FPTAS – fully polynomial time approximation scheme):

That is, for any $\epsilon > 0$, we can get a $(1 - \epsilon)$ approximation in time polynomial in N and $\frac{1}{\epsilon}$ where $N = n \times \log(\max_{i \in [n]} \{v_i, s_i\})$.

Look at a related problem (KS)

Consider a target value u , we find the min-size subset

$$\begin{array}{ll} \min_{S \subseteq [n]} & \sum_{i \in S} s_i \\ \text{s.t.} & \sum_{i \in S} v_i \geq u \end{array}$$

Let $T[i, w]$ be the min-size of subset $S \subseteq \{1, \dots, i\}$ such that $\sum_{k \in S} v_k = w$

Let $v_{\max} = \max_{i \in [n]} v_i$.

1. For $w = 0, \dots, nv_{\max}$:

$$\text{Set } T[1, w] = \begin{cases} 0 & w = 0 \\ s_1 & w = v_1 \\ \infty & \text{otherwise} \end{cases}$$

2. For $i = 2, \dots, n$:

For $w = 0, \dots, nv_{\max}$:

$$\rightarrow \text{Set } T[i, w] = \min \{T[i-1, w], T[i-1, w - v_i] + s_i\}$$

To find the optimal solution to 0-1 knapsack, it suffices to find

$$w^* = \max\{w : T[n, w] \leq B\}$$

Look at a related problem (KS)

Consider a target value u , we find the min-size subset

$$\begin{array}{ll} \min_{S \subseteq [n]} & \sum_{i \in S} s_i \\ \text{s.t.} & \sum_{i \in S} v_i \geq u \end{array}$$

Let $T[i, w]$ be the min-size of subset $S \subseteq \{1, \dots, i\}$ such that $\sum_{k \in S} v_k = w$

Let $v_{\max} = \max_{i \in [n]} v_i$.

1. For $w = 0, \dots, nv_{\max}$:

$$\text{Set } T[1, w] = \begin{cases} 0 & w = 0 \\ s_1 & w = v_1 \\ \infty & \text{otherwise} \end{cases}$$

2. For $i = 2, \dots, n$:

For $w = 0, \dots, nv_{\max}$:

$$\rightarrow \text{Set } T[i, w] = \min \{T[i-1, w], T[i-1, w - v_i] + s_i\}$$

$$T[3, w], T[3, w - 5] + 5$$

Find $w^* = \max\{w : T[n, w] \leq B\}$

Polynomial time in n and v_{\max} (but exponential in $\log v_{\max}$)!

FPTAS

We know how to solve it exactly via DP:

ExactKS $(s_1, \dots, s_n, v_1, \dots, v_n, B)$ requires $\text{poly}(n, \sum v_i)$

Goal is FPTAS: Need an algorithm that runs in $\text{poly}(n, 1/\epsilon)$
but you can relax the solution to be within $(1 - \epsilon)\mathbf{OPT}$

Algorithm: $(s_1, \dots, s_n, v_1, \dots, v_n, B, \epsilon)$

1 : $M \leftarrow \max_i v_i$

2 : $v'_i \leftarrow \left\lfloor \frac{v_i}{\epsilon M/n} \right\rfloor$

3 : $A \leftarrow \text{ExactKnapsack}(s_1, \dots, s_n, v'_1, \dots, v'_n, B)$

4 : return solution A

FPTAS

We know how to solve it exactly via DP:

ExactKS($s_1, \dots, s_n, v_1, \dots, v_n, B$) requires $\text{poly}(n, \sum v_i)$

Goal is FPTAS: Need an algorithm that runs in $\text{poly}(n, 1/\epsilon)$
but you can relax the solution to be within $(1 - \epsilon)\mathbf{OPT}$

Algorithm: ($s_1, \dots, s_n, v_1, \dots, v_n, B, \epsilon$)

1 : $M \leftarrow \max_i v_i$

2 : $v'_i \leftarrow \left\lfloor \frac{v_i}{\epsilon M/n} \right\rfloor$

3 : $A \leftarrow \text{ExactKnapsack}(s_1, \dots, s_n, v'_1, \dots, v'_n, B)$

4 : return solution A

Running time analysis:

$$\sum_{i=1}^n v'_i = \sum_{i=1}^n \left\lfloor \frac{v_i}{\epsilon M/n} \right\rfloor \leq \sum_{i=1}^n \frac{v_i}{\epsilon M/n} \leq \frac{n}{\epsilon}$$

So the running time of ExactKS is $\text{poly}\left(\frac{n^2}{\epsilon}\right) = \text{poly}\left(n, \frac{1}{\epsilon}\right)$

FPTAS

We know how to solve it exactly via DP:

ExactKS $(s_1, \dots, s_n, v_1, \dots, v_n, B)$ requires $\text{poly}(n, \sum v_i)$

Goal is FPTAS: Need an algorithm that runs in $\text{poly}(n, 1/\epsilon)$
but you can relax the solution to be within $(1 - \epsilon)\mathbf{OPT}$

Algorithm: $(s_1, \dots, s_n, v_1, \dots, v_n, B, \epsilon)$

1 : $M \leftarrow \max_i v_i$

2 : $v'_i \leftarrow \left\lfloor \frac{v_i}{\epsilon M/n} \right\rfloor$

3 : $A \leftarrow \text{ExactKnapsack}(s_1, \dots, s_n, v'_1, \dots, v'_n, B)$

4 : return solution A

Optimality analysis: Shall show $\overline{OPT} \geq (1 - \epsilon) \frac{n}{\epsilon M} \cdot OPT$. Then, we are done:

“scaled” optimal set $\sum_{i \in A} v_i \geq \frac{\epsilon M}{n} \sum_{i \in A} v'_i = \frac{\epsilon M}{n} \overline{OPT} \geq (1 - \epsilon) OPT$.

$$OPT = \sum_{i \in S} v_i = \sum_{i \in S} \frac{v_i}{\frac{\epsilon M}{n}} \frac{\epsilon M}{n} \leq \sum_{i \in S} (v'_i + 1) \frac{\epsilon M}{n} \leq \left(\sum_{i \in S} v'_i \right) \frac{\epsilon M}{n} + \epsilon M \leq \frac{\epsilon M}{n} \overline{OPT} + \epsilon OPT$$

true original optimal set