Outline

Knapsack Problem (KP)

مسئله کوله پشتی

Outline

- Introduction
- The Knapsack problem.
- A greedy algorithm for the fractional knapsack problem
- Correctness

مسئله کوله پشتی کسری

Solution $X = \langle x_1, x_2, \cdots, x_n \rangle$ دادها انتخاب

$\underline{x_i}$ مؤلفه‌های جواب

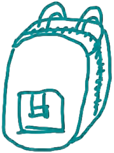
- A greedy algorithm for an optimization problem always makes the choice that looks best at the moment and adds it to the current subsolution.
- Final output is an optimal solution.
- Greedy algorithms don't always yield optimal solutions but, when they do, they're usually the simplest and most efficient algorithms available.

Outline

- Introduction
- The Knapsack problem.
- A greedy algorithm for the fractional knapsack problem
- Correctness

KP 0-1 KP

مسئله کوله‌پشتی صفر و یک



ظرفیت حمل بار

مثال

$n=3$

	A ✓	B ✓	C ✗
value	\$100	\$10	\$120
weight	$\boxed{50}$	$\boxed{5}$	$\boxed{40}$
size	2 pd	2pd	3 pd

$\frac{v_i}{w_i}$ "ارزش در واحد وزن"

← مثال نقض

حداکثر به مقدار K کیلوگرم (واحد وزن)

Capacity of knapsack: $K = 4$

$\{A, B\} \rightarrow 110\$$

$\{A, C\}$ ✗ not feasible

$\{B, C\}$ ✗ not feasible

$\{C\} \rightarrow 120\$$

بهینه

n شیء به وزن‌های $w_1, w_2, \ldots, w_n$

و ارزش‌های $v_1, v_2, \ldots, v_n$

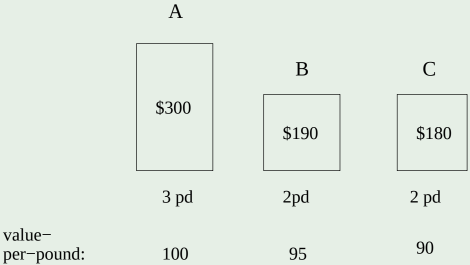
هدف: انتخاب زیرمجموعه‌ای از اشیاء که

مجموع وزن آنها از ظرفیت قابل حمل کوله‌پشتی کمتر یا مساوی باشد و مجموع ارزش‌ها بیشینه باشد

Greedy solution for 0-1 Knapsack Problem?

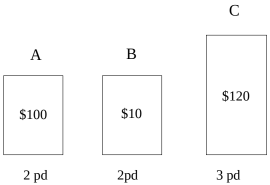
The 0-1 Knapsack Problem does not have a greedy solution!

Example



$K = 4$. Solution is item B + item C

The Knapsack Problem... FKP



Capacity of knapsack: $K = 4$

Fractional Knapsack Problem:

مسئلهٔ کوله‌پشتی کسری

شبیه مسئله کوله‌پشتی صفر و یک مطرح می‌شود
با این تفاوت که مجاز هستیم هر شیء
را بطور کامل و یا کسری از آن‌را برداریم
و در کوله‌پشتی قرار دهیم

$\Longleftarrow$ برای FKP الگوریتم حریصانه وجود دارد

The Knapsack Problem...

بطور کامل

A	B	C
$100	$10	$120
$\boxed{50}$	$\boxed{5}$	$\boxed{40}$
2 pd	2pd	3 pd

C = 2/3

$\boxed{\frac{v_i}{w_i}}$

مثال کوله پشتی کسری
(مسئله دزد و خاک طلا)

اشیاء را بر حسب ارزش در واحد وزن به ترتیب نزولی

Capacity of knapsack: $K = 4$

Fractional Knapsack Problem:
Can take a fraction of an item.

جواب بهینه برای کوله پشتی کسری

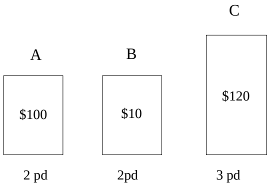
Solution:

2 pd A $100	2 pd C $80

→ 180$

0-1 Knapsack Problem: (مسئله دزد و شمش طلا)
Can only take or leave item. You can't take a fraction.

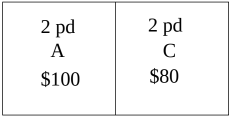
مسئله کوله پشتی صفر و یک توسط
روش برنامه ریزی پویا و نیز توسط روش پسگرد حل می شود



Capacity of knapsack: $K = 4$

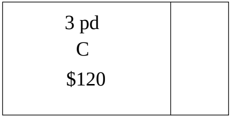
Fractional Knapsack Problem:
Can take a fraction of an item.

Solution:



0-1 Knapsack Problem:
Can only take or leave item. You can't take a fraction.

Solution:



- Given K and a set of n items:

$x_i = 0$ یعنی شیء iام انتخاب نمی‌شود

$x_i = 1$ " " بطور کامل انتخاب می‌شود

weight	w_1	w_2	...	w_n
value	v_1	v_2	...	v_n

$0 < x_i < 1$ یعنی شیء iام بطور کسری انتخاب

x_i : نشان‌دهنده انتخاب یا عدم انتخاب شیء iام

- Find: $0 \le x_i \le 1$, $i = 1, 2, \ldots, n$ such that

صورت‌بندی مسئله کوله‌پشتی در قالب یک مسئله برنامه‌ریزی خطی

(Linear Programing)

(Linear Optimization)

قیدها

Constraints

$$\boxed{\sum_{i=1}^{n} x_i w_i} \le K$$

جمع وزن اشیاء انتخاب شده

and the following is maximized:

تابع هدف

Object function

$$\max \boxed{\sum_{i=1}^{n} x_i v_i}$$

جمع ارزش اشیاء انتخاب شده

- Introduction
- The Knapsack problem.
- A greedy algorithm for the fractional knapsack problem
- Correctness

Sort items by decreasing value-per-pound

	A	B	C	D
	\$200	\$240	\$140	\$150
	1 pd	3 pd	2pd	5 pd
value−per−pound:	200	80	70	30

If knapsack holds $K = 5$ pd, solution is:

1	pd	A
3	pd	B
1	pd	C

Greedy Solution for Fractional Knapsack

- Calculate the value-per-pound $\rho_i = \frac{v_i}{w_i}$ for $i = 1, 2, \ldots, n$.

ρ_i : ارزش در واحد وزن شی iام

Greedy Solution for Fractional Knapsack

- Calculate the value-per-pound $\rho_i = \frac{v_i}{w_i}$ for $i = 1, 2, \ldots, n$.
- Sort the items by decreasing ρ_i.
  Let the sorted item sequence be $1, 2, \ldots, i, \ldots n$, and the corresponding value-per-pound and weight be ρ_i and w_i respectively.
- Let k be the current weight limit (Initially, $k = K$).
  In each iteration, we choose item i from the head of the unselected list.
  - If $k \geq w_i$, set $x_i = 1$ (we take item i), and reduce $k = k - w_i$, then consider the next unselected item.
  - If $k < w_i$, set $x_i = k/w_i$ (we take a fraction k/w_i of item i), Then the algorithm terminates.

1 • Calculate the value-per-pound $\rho_i = \frac{v_i}{w_i}$ for $i = 1, 2, \ldots, n$.

2 • Sort the items by decreasing ρ_i.
Let the sorted item sequence be $1, 2, \ldots, i, \ldots n$, and the corresponding value-per-pound and weight be ρ_i and w_i respectively.

k ظرفیت فضای خالی مانده

3 • Let k be the current weight limit (Initially, $k = K$).
In each iteration, we "choose" item i from the head of the unselected list.

- a • If $k \geq w_i$, set $x_i = 1$ (we take item i), and reduce $k = k - w_i$, then consider the next unselected item.
- b • If $k < w_i$, set $x_i = k/w_i$ (we take a fraction k/w_i of item i), Then the algorithm terminates.

Running time: $O(n \log n)$.

- Observe that the algorithm may take a fraction of an item. This can only be the last selected item.

Greedy Solution for Fractional Knapsack

- Observe that the algorithm may take a fraction of an item. This can only be the last selected item.
- We claim that the total value for this set of items is the optimal value.

Given a set of n items $\{1, 2, ..., n\}$.

- Assume items sorted by per-pound values: $\rho_1 \geq \rho_2 \geq ... \geq \rho_n$.

Let the greedy solution be $G = \langle x_1, x_2, ..., x_k \rangle$

- x_i indicates fraction of item i taken (all $x_i = 1$, except possibly for $i = k$).

Consider any optimal solution $O = \langle y_1, y_2, ..., y_n \rangle$

- y_i indicates fraction of item i taken in O $\big($for all i, $0 \leq y_i \leq 1\big)$.
- Knapsack must be full in both G and O:
  $\sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} y_i w_i = K$.

Consider the first item i where the two selections differ.

- By definition, solution G takes a greater amount of item i than solution O

اصطلاحاً می‌گوییم مسئله KP جواب بدیهی دارد
اگر $\sum w_i < K$ باشد یعنی تمام اشیاء را می‌توان برداشت

Given a set of n items $\{1, 2, ..., n\}$.

- Assume items sorted by per-pound values: $\rho_1 \geq \rho_2 \geq ... \geq \rho_n$.

Let the greedy solution be $G = \langle x_1, x_2, ..., x_k \rangle, x_{k+1} = 0, \cdots, x_n = 0\rangle$

- x_i indicates fraction of item i taken (all $x_i = 1$, except possibly for $i = k$).

Consider any optimal solution $O = \langle y_1, y_2, ..., y_n \rangle$

- y_i indicates fraction of item i taken in O (for all i, $0 \leq y_i \leq 1$).
- Knapsack must be full in both G and O:
  $\sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} y_i w_i = K$.

(تحت این فرض که مسئله جواب بدیهی نداشته باشد)

Consider the first item i where the two selections differ.

- By definition, solution G takes a greater amount of item i than solution O (because the greedy solution always takes as much as it can).

Correctness

Given a set of n items $\{1, 2, ..., n\}$.

- Assume items sorted by per-pound values: $\rho_1 \geq \rho_2 \geq ... \geq \rho_n$.

Let the greedy solution be $G = \langle x_1, x_2, ..., x_k \rangle$

- x_i indicates fraction of item i taken (all $x_i = 1$, except possibly for $i = k$).

Consider any optimal solution $O = \langle y_1, y_2, ..., y_n \rangle$

- y_i indicates fraction of item i taken in O $\left(\text{for all } i,\ 0 \leq y_i \leq 1\right)$.
- Knapsack must be full in both G and O: $\sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} y_i w_i = K$.

Consider the first item i where the two selections differ.

- By definition, solution G takes a greater amount of item i than solution O (because the greedy solution always takes as much as it can). Let $x = x_i - y_i$.

$x := x_i - y_i$

Consider the following new solution O' constructed from O:

- For $j < i$, keep $y'_j = y_j$.
- Set $y'_i = x_i$.
- In O, remove items of total weight xw_i from items $i+1$ to n, resetting the y'_j appropriately.

$O' = \langle y_1, \cdots, y_{i-1}, x_i, y'_{i+1}, \cdots, y'_n \rangle$

مقدار وزن اضافی

  This is always doable because $\sum_{j=i}^{n} x_j = \sum_{j=i}^{n} y_j$ ✓✓

- The total value of solution O' is greater than or equal to the total value of solution O (why?)
- Since O is largest possible solution and value of O' cannot be smaller than that of O, O and O' must be equal.
- Thus solution O' is also optimal.

By repeating this process, we will eventually convert O into G, without changing the total value of the selection.

Therefore G is also optimal!

جواب دیگری مثل O' از روی O می‌سازیم

- از شیء اول تا شیء $(i-1)$ همانند قبل برمی‌داریم

- شیء iام را دقیقاً به مقدار x_i برمی‌داریم؛

نتیجتاً در مقایسه با جواب قبلی O، وزنی به مقدار $\delta \cdot w_i$ اضافی برداشته‌ایم

$$\delta := x_i - y_i$$

- از هر یک از اشیاء $(i+1)$ام تا nام قدری می‌کاهیم؛

به گونه‌ای که مجموع این کاهش برابر با مقدار اضافی، $\delta \cdot w_i$ شود.

حال مجموع ارزش جواب جدید O' را با ارزش O مقایسه می‌کنیم:

$$\text{ارزش } O' = \sum_{j=1}^{i-1} y_j v_j + x_i v_i + \underbrace{\sum_{j=i+1}^{n} y'_j v_j}$$

$$\underbrace{\sum_{j=i+1}^{n} y'_j v_j} = \sum_{j=i+1}^{n} (y_j - \varepsilon_j) v_j$$

$$= \sum_{j=i+1}^{n} y_j v_j - \sum_{j=i+1}^{n} \varepsilon_j \cdot w_j \cdot \frac{v_j}{w_j}$$

$$\geq \sum_{j=i+1}^{n} y_j v_j - \frac{v_i}{w_i} \sum_{j=i+1}^{n} \varepsilon_j \cdot w_j$$

$$\Rightarrow \text{ارزش } O' \geq \sum_{j=1}^{i-1} y_j v_j + x_i v_i + \sum_{j=i+1}^{n} y_j v_j - \frac{v_i}{w_i} \underbrace{\sum_{j=i+1}^{n} \varepsilon_j \cdot w_j}_{= \delta \cdot w_i}$$

مجموع وزنی که به شیء iام اضافه و از اشیاء بعدی کاسته‌ام

$$\Rightarrow \text{ارزش } O' \geq \sum_{j=1}^{i-1} y_j v_j + (y_i + \delta) v_i + \sum_{j=i+1}^{n} y_j v_j - \frac{v_i}{\cancel{w_i}} \cdot \delta \cdot \cancel{w_i}$$

$$= \sum_{j=1}^{i-1} y_j v_j + y_i v_i + \sum_{j=i+1}^{n} y_j v_j$$

$$= \text{ارزش } O$$

$$\Rightarrow \text{ارزش } O' \geq \text{ارزش } O \Rightarrow$$

۱) یا O' جواب بهتری است ← تناقض

۲) یا O' نیز یک جواب بهینه است اما تا شیء iام همانند جواب حریصانه است ← تناقض