

$$\begin{array}{ll}\max & f(x) \\ \text{subject to} & x \in \mathcal{S},\end{array}$$

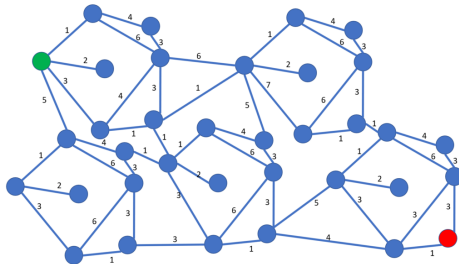
where:

- $f(x)$  is an objective function
- $\mathcal{S}$  is a feasible set
- $x \in \mathcal{S}$  is a feasible solution

$$\begin{aligned} \max \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{S}, \end{aligned}$$

where:

- $f(x)$  is an objective function
- $\mathcal{S}$  is a feasible set
- $x \in \mathcal{S}$  is a feasible solution



Shortest path problem: Find a path of minimum distance from the red node to the green node?

A feasible path  $x = (v_0, v_1, v_2, \dots, v_m, v_{m+1})$

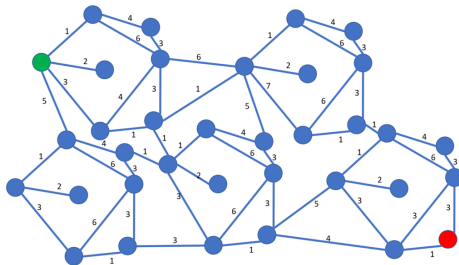
A feasible set  $\mathcal{S} = \{x \mid x \text{ is a feasible path}\}$

Objective value  $f(x) = \sum_{i=0}^m w_{i,i+1}$

$$\begin{aligned} \max \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{S}, \end{aligned}$$

where:

- $f(x)$  is an objective function
- $\mathcal{S}$  is a feasible set
- $x \in \mathcal{S}$  is a feasible solution



Shortest path problem: Find a path of minimum distance from the red node to the green node?

A feasible path  $x = (v_0, v_1, v_2, \dots, v_m, v_{m+1})$

A feasible set  $\mathcal{S} = \{x \mid x \text{ is a feasible path}\}$

Objective value  $f(x) = \sum_{i=0}^m w_{i,i+1}$

Exhaustive search (or brute-force search) can find a shortest path but is time-consuming!

## Integer Linear Programming

$$\begin{array}{ll}\max & c^T x \\ \text{subject to} & Ax \leq b, \\ & x \in \{0, 1\}^n,\end{array}$$

where

- $x = (x_1, \dots, x_n)$  is a vector of variables
- $A$  is an  $m \times n$  matrix
- $b$  is an  $m$ -dimensional vector
- $c$  is an  $n$ -dimensional vector

Many discrete optimization problems can be formulated as a Integer Linear Program

- Knapsack, Shortest path, TSP, ...

## Model

$$\begin{aligned} \max \quad & \sum_{i=1}^n c_i x_i \\ \text{subject to} \quad & \sum_{i=1}^n w_i x_i \leq W, \\ & x \in \{0, 1\}^n, \end{aligned}$$

where

- $c_i$  is the profit of item  $i$ th
- $w_i$  is the weight of item  $i$ th
- $B$  is the capacity

## How to solve Knapsack

- Brute-force search takes  $O(2^n)$  times in the worst case

## Model

$$\begin{aligned} \max \quad & \sum_{i=1}^n c_i x_i \\ \text{subject to} \quad & \sum_{i=1}^n w_i x_i \leq W, \\ & x \in \{0, 1\}^n, \end{aligned}$$

where

- $c_i$  is the profit of item  $i$ th
- $w_i$  is the weight of item  $i$ th
- $B$  is the capacity

## How to solve Knapsack

- Brute-force search takes  $O(2^n)$  times in the worst case
- Is there any algorithm better?

## Model

$$\begin{aligned} \max \quad & \sum_{i=1}^n c_i x_i \\ \text{subject to} \quad & \sum_{i=1}^n w_i x_i \leq W, \\ & x \in \{0, 1\}^n, \end{aligned}$$

where

- $c_i$  is the profit of item  $i$ th
- $w_i$  is the weight of item  $i$ th
- $B$  is the capacity

## How to solve Knapsack

- Brute-force search takes  $O(2^n)$  times in the worst case
- Is there any algorithm better? **Yes: Dynamic Programming**

## Ý tưởng

- Chia bài toán thành các bài toán con (có kích thước nhỏ hơn)
- Tìm mối liên hệ giữa lời giải tối ưu bài toán gốc với lời giải của các bài toán
- Tìm lời giải cho các bài toán con, và lưu kết quả tìm được dưới dạng bảng
- Tìm lời giải tối ưu từ các lời giải có trong bảng



Xây dựng mảng hai chiều  $V = [0..n, 0..W]$

- Với  $1 \leq i \leq n$ , và  $0 \leq w \leq W$ ,  $V[i, w]$  là giá trị tối ưu của bài toán con với  $i$  đồ vật đầu tiên  $1, 2, \dots, i$  và  $W = w$ .
- $V[n, W]$  là giá trị tối ưu của bài toán gốc ban đầu

Xây dựng mảng hai chiều  $V = [0..n, 0..W]$

- Với  $1 \leq i \leq n$ , và  $0 \leq w \leq W$ ,  $V[i, w]$  là giá trị tối ưu của bài toán con với  $i$  đồ vật đầu tiên  $1, 2, \dots, i$  và  $W = w$ .
- $V[n, W]$  là giá trị tối ưu của bài toán gốc ban đầu

Mối liên hệ giữa lời giải của một bài toán với lời giải của các bài toán con của nó

Khởi tạo:

$$\begin{aligned} V[0, w] &= 0 & \text{nếu } 0 \leq w \leq W, \\ V[i, w] &= -\infty & \text{nếu } w < 0. \end{aligned}$$

Công thức đệ quy:

$$V[i, w] = \max\{V[i-1, w], c_i + V[i-1, w - w_i]\}$$

với mọi  $1 \leq i \leq n$ ,  $0 \leq w \leq W$ .

## Công thức đệ quy

$$V[i, w] = \max\{V[i-1, w], c_i + V[i-1, w - w_i]\} \quad (*)$$

với mọi  $1 \leq i \leq n$ ,  $0 \leq w \leq W$ .

### Chứng minh:

Giả sử công thức (\*) đúng với mọi  $1 \leq i \leq i^* - 1$ ,  $0 \leq w \leq W$ . Ta phải chứng minh nó đúng với  $i = i^*$ , tức là:

$$V[i^*, w] = \max\{V[i^* - 1, w], c_{i^*} + V[i^* - 1, w - w_{i^*}]\}$$

## Công thức đệ quy

$$V[i, w] = \max\{V[i-1, w], c_i + V[i-1, w - w_i]\} \quad (*)$$

với mọi  $1 \leq i \leq n$ ,  $0 \leq w \leq W$ .

**Chứng minh:**

Giả sử công thức (\*) đúng với mọi  $1 \leq i \leq i^* - 1$ ,  $0 \leq w \leq W$ . Ta phải chứng minh nó đúng với  $i = i^*$ , tức là:

$$V[i^*, w] = \max\{V[i^* - 1, w], c_{i^*} + V[i^* - 1, w - w_{i^*}]\}$$

Chú ý rằng để tính  $V[i^*, w]$  có hai phương án:

- Không chọn đồ vật  $i^*$ : Rõ ràng

$$V[i^*, w] = V[i^* - 1, w]$$

## Công thức đệ quy

$$V[i, w] = \max\{V[i-1, w], c_i + V[i-1, w - w_i]\} \quad (*)$$

với mọi  $1 \leq i \leq n$ ,  $0 \leq w \leq W$ .

**Chứng minh:**

Giả sử công thức (\*) đúng với mọi  $1 \leq i \leq i^* - 1$ ,  $0 \leq w \leq W$ . Ta phải chứng minh nó đúng với  $i = i^*$ , tức là:

$$V[i^*, w] = \max\{V[i^* - 1, w], c_{i^*} + V[i^* - 1, w - w_{i^*}]\}$$

Chú ý rằng để tính  $V[i^*, w]$  có hai phương án:

- **Không chọn đồ vật  $i^*$ :** Rõ ràng

$$V[i^*, w] = V[i^* - 1, w]$$

- **Chọn đồ vật  $i^*$ :** (chỉ xảy ra nếu  $w_i \leq w$ ):

$$V[i^*, w] = c_{i^*} + V[i^* - 1, w - w_{i^*}]$$

Thứ tự xác định các phần tử của ma trận  $V[i, w]$ .

$V[i, w]$	$w=0$	1	2	3	...	...	$W$	
$i=0$	0	0	0	0	...	...	0	bottom
1								
2								
$\vdots$								
$n$								up

Thời gian chạy của thuật toán:  $O(nW)$ .

Ví dụ: Xét bài toán với  $W = 10$  và:

$i$	1	2	3	4
$c_i$	10	40	30	50
$w_i$	5	4	6	3

Ví dụ: Xét bài toán với  $W = 10$  và:

$i$	1	2	3	4
$c_i$	10	40	30	50
$w_i$	5	4	6	3

$V[i, w]$	0	1	2	3	4	5	6	7	8	9	10
$i = 0$	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	10	10	10	10	10	10
2	0	0	0	0	40	40	40	40	40	50	50
3	0	0	0	0	40	40	40	40	40	50	70
4	0	0	0	50	50	50	50	90	90	90	90

Kết luận: giá trị tối ưu:  $V[4, 10] = 90$ .

Câu hỏi: Tìm phương án tương ứng với lời giải tối ưu thế nào?



Ví dụ: Xét bài toán với  $W = 5$  và:

$i$	1	2	3	4
$c_i$	10	15	25	12
$w_i$	1	2	3	2

Ví dụ: Xét bài toán với  $W = 50$  và:

$i$	1	2	3	4
$c_i$	2	3	4	1
$w_i$	10	40	20	30

Gọi  $P = \sum_{i=1}^n c_i$ .

Xây dựng mảng hai chiều  $V = [0..n, 0..P]$

- Với  $1 \leq i \leq n$ , và  $0 \leq c \leq P$ ,  $V[i, c]$  là trọng lượng nhỏ nhất có thể của tập con của  $i$  đồ vật đầu tiên  $1, 2, \dots, i$  sao cho tổng giá trị của tập con đó bằng đúng  $p$ .
- $V[n, P]$  là giá trị tối ưu của bài toán gốc ban đầu

Mối liên hệ giữa lời giải của một bài toán với lời giải của các bài toán con của nó

Khởi tạo:

$$\begin{aligned} V[0, 0] &= 0, V[0, p] = W + 1 && \text{với } 1 \leq p \leq P, \\ V[i, p] &= +\infty && \text{với } p < 0. \end{aligned}$$

Công thức đệ quy:

$$W[i, p] = \min\{V[i-1, p], w_i + V[i-1, p - c_i]\}$$

với mọi  $1 \leq i \leq n$ ,  $0 \leq p \leq P$ .

### Subset sum

Cho tập  $S = \{a_1, a_2, \dots, a_n\}$  gồm các số nguyên dương, và một số  $W > 0$ . Hỏi rằng liệu có tồn tại tập con  $S'$  của  $S$  sao cho tổng các phần tử trong  $S'$  đúng bằng  $W$ .

### Partition

Cho tập  $S = \{a_1, a_2, \dots, a_n\}$  gồm các số nguyên dương. Hỏi rằng liệu có tồn tại tập con  $S'$  của  $S$  sao cho tổng các phần tử trong  $S'$  đúng bằng tổng các phần tử trong tập  $S \setminus S'$ .