Lezione 11

11.1 DIMOSTRAZIONE DELLE PROPRIETÀ DI SOTTOSTRUTTURA OTTIMA E DI SCELTA GREEDY NEI PROBLEMI DELLO ZAINO

KNAPSACK (0-1 KNAPSACK O KNAPSACK INTERO)

input:
- n oggetti tali che per ogni oggetto i sia specificato un peso w_i e un valore v_i, $\forall i \in \{1, \dots, n\}$;
- un limite di peso W.

goal: scegliere un sottoinsieme $J \subseteq \{1, \dots, n\}$ di oggetti di valore complessivo massimo e peso complessivo limitato superiormente da W.

oppure (equivalentemente)

trovare un assegnamento per le variabili $\{x_1, \dots, x_n\}$ (ognuna corrisponde ad un oggetto, $x=0$ non prendo l'oggetto, $x=1$ prendo l'oggetto)

$$
\begin{cases}
\max \sum_{i=1}^{n} v_i x_i & \leftarrow \text{soft constraints} \Leftrightarrow \text{preferenze/funzione di costo} \\
\sum_{i=1}^{n} w_i x_i \le W \\
x_i \in \{0,1\}, \ \forall i \in \{1, \dots, n\}
\end{cases}
$$

requisiti stretti $\Updownarrow$ hard constraints

FRACTIONAL KNAPSACK

<u>input</u>:
- n oggetti tali che per ogni oggetto i sia specificato un peso w_i e un valore v_i, $\forall i \in \{1, \dots, n\}$;
- un limite di peso W.

<u>goal</u>: trovare un assegnamento per le variabili $\{x_1, \dots, x_n\}$ tale che

$$\begin{cases} \max \sum_{i=1}^{n} v_i x_i & \text{] funzione obiettivo fatta di funzioni di costo} \\ \sum_{i=1}^{n} w_i x_i \le W & \\ 0 \le x_i \le 1 \,,\; x_i \in \mathbb{Q} \quad \forall i \in \{1, \dots, n\} & \end{cases}$$

requisiti stretti

PROP: $\{0,1\}$-KNAPSACK HA LA PROPRIETÀ DI SOTTOSTRUTTURA OTTIMA.

<u>Proof</u>: Sia $\{x_1^*, x_2^*, \dots, x_n^*\}$ "S^*" una soluzione ottima al problema allora $\sum_{i=1}^{n} w_i x_i^* \le W$ e $\sum_{i=1}^{n} v_i x_i^*$ è massimo.

Consideriamo il sottoproblema le cui variabili sono

$$\{x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n\}_{S^* \text{ ristretto}}$$

Supponiamo che la restrizione $\{x_1^*, \dots, x_{j-1}^*, x_{j+1}^*, \dots, x_n^*\}$ non sia ottima per questo sottoproblema,

$\Rightarrow \exists\, S' = \{x_1', \dots, x_{j-1}', x_{j+1}', \dots, x_n'\}$ soluzione tale che

$$\boxed{\sum_{\substack{i=1 \\ i \ne j}}^{n} v_i x_i' > \sum_{\substack{i=1 \\ i \ne j}}^{n} v_i x_i^*} \,, \quad \sum_{\substack{i=1 \\ i \ne j}}^{n} w_i x_i' \le W - x_j^* w_j$$

allora $\{x_1', x_2', \dots, x_{j-1}', x_j^*, x_{j+1}', \dots, x_n'\}$ è una soluzione al problema originario migliore di $\{x_1^*, \dots, x_j^*, \dots, x_n^*\}$

perché

$$\sum_{\substack{i=1\\i\neq j}}^{n} v_i x_i' + v_j x_j^* > \sum_{\substack{i=1\\i\neq j}}^{n} v_i x_i^* + v_j x_j^* = \sum_{i=1}^{n} v_i x_i^*$$

assurdo perché è ottima.

$$\underbrace{\sum_{\substack{i=1\\i\neq j}}^{n} w_i x_i'}\ + w_j x_j^* \leq W - w_j x_j^* + w_j x_j^* = W \ . \quad \square$$

Non abbiamo usato da nessuna parte $x_i^* \in \{0,1\}$, quindi la stessa dimostrazione può essere ripetuta per dimostrare che

PROP: FRACTIONAL KNAPSACK HA LA PROPRIETÀ DI SOTTOSTRUTTURA OTTIMA.

Allora per entrambi i problemi posso usare la programmazione dinamica.

Possiamo usare la strategia greedy?

Euristica che possiamo usare: scegliere l'oggetto con più grande valore per unità di peso

per ogni oggetto $k_i = \frac{v_i}{w_i}$ più grande.

prendo l'oggetto con

$\leadsto$ otteniamo un algoritmo in $O(n \log n)$-time

PROP: FRACTIONAL KNAPSACK HA LA PROPRIETÀ DI SCELTA GREEDY.

PROOF: $k_1 \geq \dots \geq k_n$.

$\Leftrightarrow$
$K_1 \geq K_2$

$$\sum_{i=1}^{n} x_i v_i = \sum_{i=1}^{n} q_i k_i, \qquad \sum_{i=1}^{n} q_i \leq W, \quad q_i = x_i w_i$$

Sia $\{q_1, \dots, q_n\}$ una soluzione ottima che non contiene K_1 (la scelta greedy). $\rightsquigarrow q_1 = 0$ $\rightarrow \{q_1, q_2, q_3, \dots, q_n\}$

Allora supponiamo $q_2 \neq 0$. Allora possiamo sostituire il peso dell'oggetto 2 con lo stesso peso dell'oggetto 1, ottenendo

$$\{q_1' = q_2, \; q_2' = 0, \; q_3' = q_3^*, \dots, q_n' = q_n^*\}$$

allora

(il peso) $$\sum_{i=1}^{n} q_i' = q_1' + q_2' + \sum_{i=3}^{n} q_i = q_2 + \underset{\substack{\| \\ q_1}}{0} + \sum_{i=3}^{n} q_i = \sum_{i=1}^{n} q_i \leq W$$

(il valore) $$\sum_{i=1}^{n} q_i' k_i = q_1' K_1 + q_2' K_2 + \sum_{i=3}^{n} q_i k_i = q_2 K_1 + 0 \cdot K_2 + \sum_{i=3}^{n} q_i k_i \geq q_2 K_2 + \sum_{i=3}^{n} q_i k_i = \underbrace{\sum_{i=1}^{n} q_i k_i}_{\text{ottimo}}$$

$\Rightarrow \sum_{i=1}^{n} q_i' K_i$ è massimo.

Quindi ho trovato una soluzione ottima che contiene la scelta greedy. $\square$

⇒ Posso usare la strategia greedy per fractional Knapsack.

E per $\{0,1\}$-Knapsack?

$\{0,1\}$-KNAPSACK <u>NON</u> HA LA PROPRIETÀ DI SCELTA GREEDY.

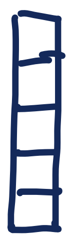
<u>CONTROESEMPIO:</u> Supponiamo di avere 3 oggetti

$i=1$, $w_1 = 10$ kg, $v_1 = 60$ €, $k_1 = 6$ €/kg

$i=2$, $w_2 = 20$ kg, $v_2 = 100$ €, $k_2 = 5$ €/kg

$i=3$, $w_3 = 30$ kg, $v_3 = 120$ €, $k_3 = 4$ €/kg

$W = 50$ kg



Possibili modi per riempire lo zaino (soluzioni)

⇒ Non esiste una sol ottima che contenga le scelte greedy.

Se avessimo ammesso soluzioni frazionarie

$\frac{2}{3}$ ogg 3 $\rightsquigarrow x_3 = \frac{2}{3}$

ogg. 2 $\rightsquigarrow x_2 = 1$

oggetto 1 $\rightarrow x_1 = 1$

Valore: $160 + \frac{2}{3} \cdot 120 = 160 + 80 = 240$ € $>$ 220 €.

11.2 RICERCA DEPTH FIRST

Sia $G = (V, E)$ un grafo (orientato).

La ricerca Depth-First Search (DFS) esplora gli archi che dipartono dal vertice v scoperto più recentemente e che ancora sono inesplorati. Una volta che tutti gli archi uscenti da v sono stati esplorati, la ricerca fa retromarcia per esplorare gli archi uscenti dal vertice u da cui aveva raggiunto v.

Il processo continua finché non tutti i vertici raggiungibili dal vertice iniziale sorgente sono stati esplorati.

Se nel grafo ci sono ancora dei vertici non esplorati la ricerca DFS seleziona un nuovo vertice sorgente e ripete la procedura.

Grafo <u>dei predecessori</u> è una foresta che comprende uno o più alberi dei predecessori e che rappresenta l'esplorazione fatta.

$$G_\pi = (V, E_\pi) \quad \text{con } E_\pi = \{(v.\pi, v) : v \in V, v.\pi \neq NIL\}$$