INAPPROSSIMABILITÀ DEL TSP GENERALE

Decidere se un grafo contiene un circuito hamiltoniano è NP-completo.

Teorema: Non esiste alcun $\alpha \geq 1$ t.c. TSP sia α-approssimabile, a meno che $P = NP$.

Dim: per assurdo

$$G = (V, E)$$

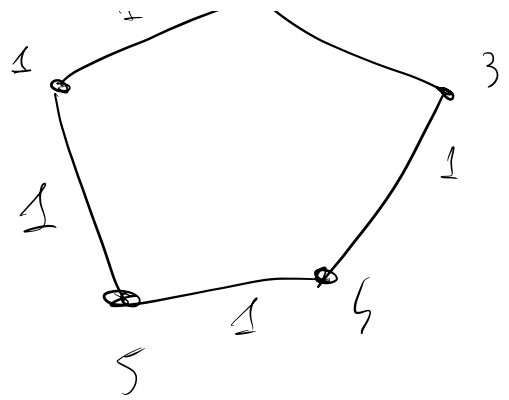
$$G' = \left(V, \binom{V}{2}\right) \quad \langle d_e \rangle$$

$$d(x,y) = \begin{cases} 1 & \text{se } xy \in E \\ \lceil \alpha n \rceil + 1 & \text{altrim.} \end{cases}$$

G ammette un circuito hamiltoniano

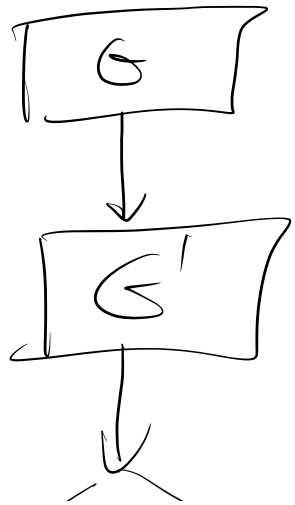
$\Rightarrow$ in G' quel circuito ha costo n

1 2 1



G non ammette un circuito hamiltoniano $\Rightarrow$ in G' tutti i circuiti hamiltoniani passano per un lato di costo $\lceil \alpha n \rceil + 1$

$\Rightarrow$ in G' il costo del circ. hamiltoniano minimo è $\geq \lceil \alpha n \rceil + 1$



Algoritmo α approx. per TSP

G aveva un circuito hamilt. ⇓ c'ne ha uno di peso n

trova un circ. hamiltoniano di peso $\leq \alpha n$

G non ha. ⇓ c'ne ha solo di peso $\geq \lceil \alpha n \rceil + 1$

trova un circ. hamiltoniano di peso $\geq \lceil \alpha n \rceil + 1$

È impossibile che

$$\alpha n < \lceil \alpha n \rceil + 1$$

cioè deve essere

$$\alpha n \geq \lceil \alpha n \rceil + 1$$

$$\alpha \geq \frac{\lceil \alpha n \rceil + 1}{n} \geq \frac{\alpha n + 1}{n} = \alpha + \frac{1}{n}$$

$$\alpha \geq \alpha + \frac{1}{n}$$

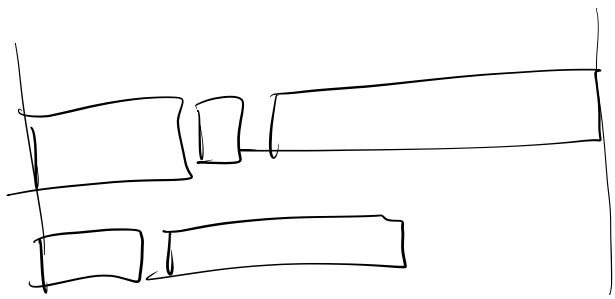
IMPOSSIBILE. $\square$

$$TSP \notin APX$$

$$\text{Metric-}TSP \in \frac{3}{2}\text{-}APX \subseteq APX$$

UN PTAS PER 2-LOAD BALANCING

INPUT: $t_0, t_1, \dots, t_{n-1} \in \mathbb{N}^+$



ALGORITMO

INPUT: $t_0, t_1, \dots, t_{n-1} \in \mathbb{N}^+$

$\varepsilon > 0$

- IF $\varepsilon \geq 1$
  - assegna tutti i task alla prima macchina
- ALTRIMENTI
  - ordina i task

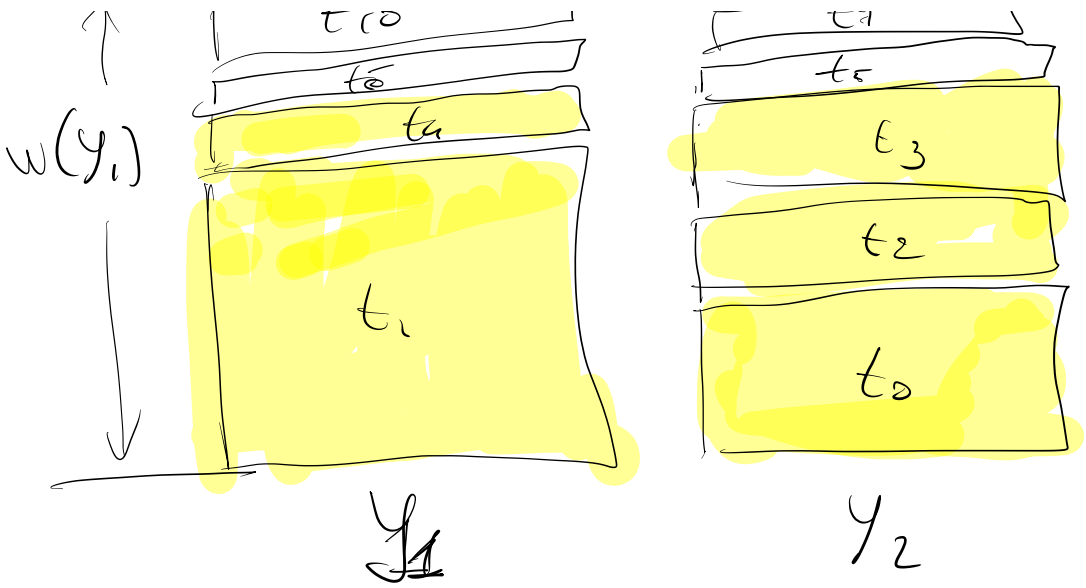
    $t_0 \geq t_1 \geq \dots \geq t_{n-1}$

- sia
$$k \leftarrow \left\lceil \frac{1}{\varepsilon} - 1 \right\rceil$$
- cerca la partizione ottima per $\{t_0, t_1, \dots, t_{k-1}\}$
- Assegna i task restanti $t_k, t_{k+1}, \dots, t_{n-1}$ come nell'algoritmo greedy

<u>Teorema</u>: Questo algoritmo è polinomiale in n e produce una $(1+\varepsilon)$-appross.

<u>Dim</u>: Se $\varepsilon \geq 1$, ovvio.

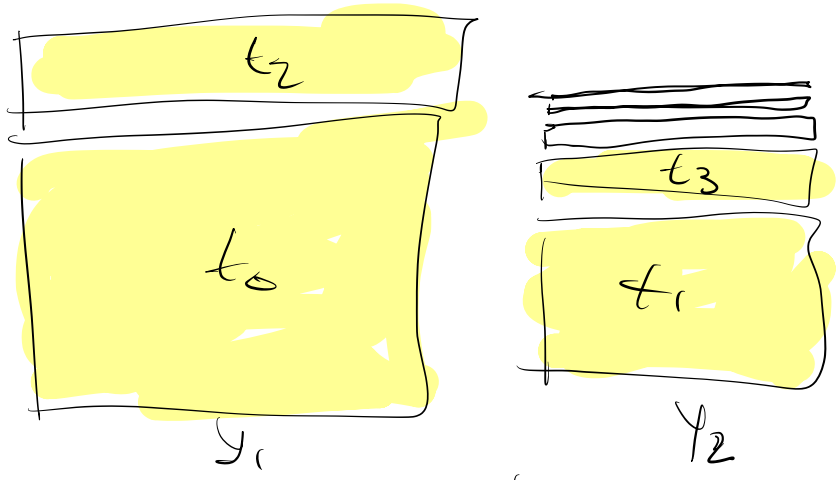
$$\underbrace{t_0 \geq t_1 \geq \dots \geq t_{k-1}}_{k \text{ task}} \geq \underline{t_k \geq \dots \geq t_{n-1}}$$



S.p.g. $w(Y_1) \geq w(Y_2)$

Sia h l'indice dell'ultimo task assegnato a Y_1.

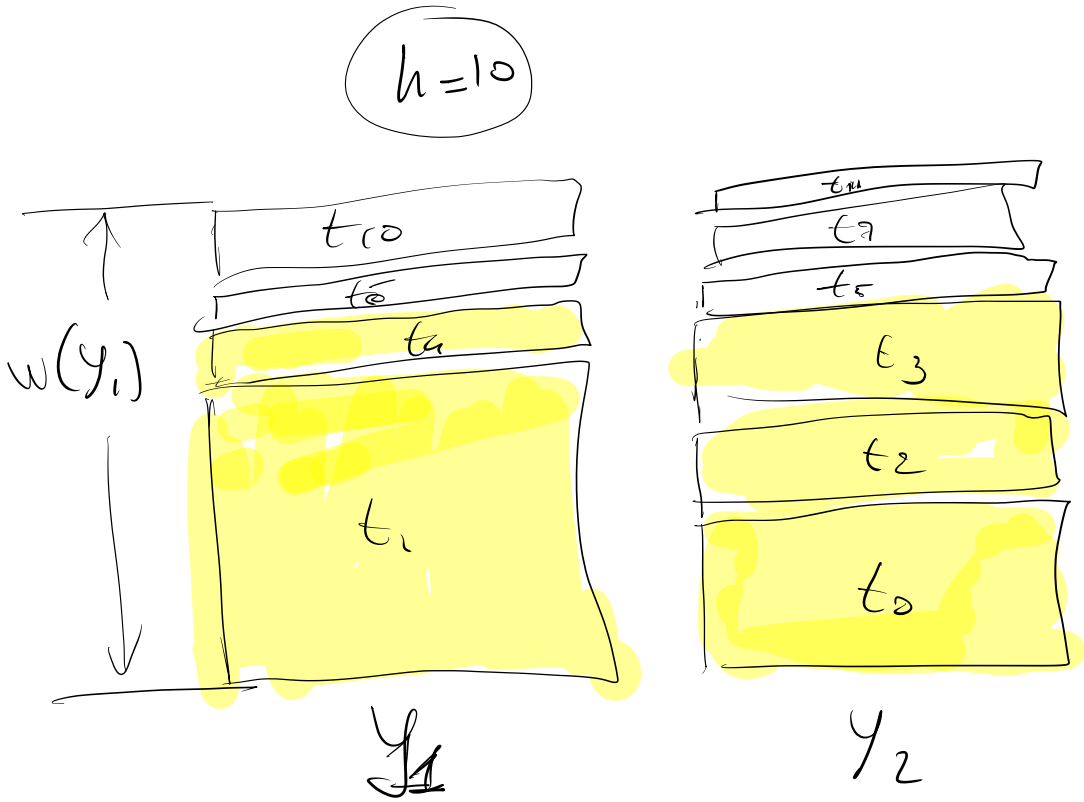
$\boxed{\text{CASO 1 : } h < k}$



In questo caso la soluzione

e ottima.

$\boxed{\text{CASO 2: } h \geq R}$



$w(Y_1) - t_h \leq$ carico di Y_2 nel momento in cui abbiamo assegnato $t_h \leq w(Y_2)$

$$2w(Y_1) - t_h \leq \underbrace{w(Y_1) + w(Y_2)}_{= 2L} =$$

$$\boxed{L \triangleq \frac{t_0 + t_1 + \dots + t_{n-1}}{2}}$$

$$\boxed{w(y_1) - \frac{t_h}{2} \le L} \qquad (*)$$

$$2L = \underbrace{\boxed{t_0 + t_1 + \dots}}_{t_k} + \underbrace{t_{n-1}}_{t_h} \ge$$

$$\ge \overbrace{\underbrace{t_h + t_h \dots + t_h}}^{k} + \underbrace{t_k + t_{k+1} \dots t_{n-1}}_{2}$$

$$\ge t_h (k+1) \qquad (**)$$

$$\frac{w(y_1)}{w^*} \underset{w^* \ge L}{\le} \frac{w(y_1)}{L} \underset{(*)}{\le} \frac{\frac{t_h}{2} + L}{L} =$$

$$= 1 + \frac{t_h}{2L} \le 1 + \frac{\cancel{t_h}}{\cancel{t_h}(k+1)} \le$$

$k \geq \frac{1}{\varepsilon} - 1$

$k+1 \geq \frac{1}{\varepsilon}$

$$\leq 1 + \frac{1}{1+k} \overset{(**)}{\leq} 1 + \frac{1}{1/\varepsilon} = 1 + \varepsilon$$

$\square$

<u>OSSERVAZIONE</u>: L'algoritmo ha tempo d'esecuzione $O(n \lg n + 2^{\frac{1}{\varepsilon}})$

$$k = \left\lceil \frac{1}{\varepsilon} - 1 \right\rceil \leq \frac{1}{\varepsilon}$$

$$2^k = 2^{\frac{1}{\varepsilon}}$$

$\square$

KNAPSACK o ZAINO

INPUT: $v_0, v_1, \dots, v_{n-1} \in \mathbb{Q}^+$

$w_0, w_1, \dots, w_{n-1}$

$W \in \mathbb{Q}^+$

SOL. AMMISS.: $X \subseteq \mathcal{N}$

t.c. $\sum_{i \in X} w_i \leq W$

FUNZ. OB.: $\sum_{i \in X} v_i$

TIPO: MAX

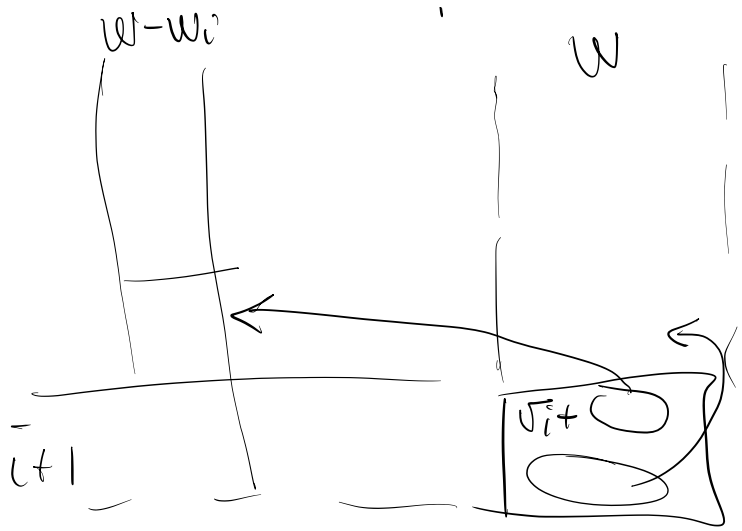
Teorema: KNAPSACK $\in$ NPO - Completo

1° ALG. PROGRAMMAZIONE DINAMICA

$i \leq n$ $\quad$ $w \leq W$

$v\text{-}OPT[i, w] \triangleq$ MASSIMO VALORE OTTENIBILE CON I PRIMI i OGGETTI E AVENDO UNO ZAINO DI CAPIENZA w

	$w=0$	$w=1$	...	$w=W$
$i=0$	0	0		0
$i=1$				
$i=2$				
$\vdots$				
$i=n$				



$v_i \neq w_i$

$$\boxed{vOPT[i+1, w] = \max\left(vOPT[i, w],\ vOPT[i, w - w_i] + v_i\right)}$$