EXERCISE 4 Show that if $P \neq NP$ there cannot exist an FPTAS for Δ_TSP

trovare NPC problema risolvibile tramite FPTAS

Contraddizione: esiste $\Rightarrow A_{\Delta TSP}(\langle G, w\rangle, \varepsilon) \Rightarrow$ considero HAMILTON

$(I: \langle G=(V,E)\rangle, Q: \exists$ ciclo hamiltoniano?$)$

$\langle G=(V,E)\rangle \rightsquigarrow \langle G_c=(V,E_c), w\rangle, \quad w=\begin{cases}1 & \text{if } (u,v)\in E \\ 2 & \text{otherwise}\end{cases} \Rightarrow$ soddisfa Δ

Considero $\langle G\rangle \in$ HAMILTON $\Rightarrow G_c$ ha ciclo di costo $|V|$

$A_{\Delta TSP}(\langle G_c, w\rangle, \varepsilon)$ ritorna $\hat{T} \Rightarrow w(\hat{T}) = (1+\varepsilon)|V| \Rightarrow$ scelgo $\varepsilon : w(\hat{T}) < |V|+1 \Rightarrow$

$\Rightarrow \varepsilon = \frac{1}{2|V|}$ (ε preso a piacere dato vincolo) $\Rightarrow w(\hat{T}) \leq \left(1+\frac{1}{2|V|}\right)|V| = |V| + \frac{1}{2} \Rightarrow$

$\Rightarrow$ costi tutti interi: $w(\hat{T}) \leq |V| \Rightarrow A_{\Delta TSP}(\langle G_c, w\rangle, \frac{1}{2|V|})$ ritorna ciclo hamiltoniano

Considero $\langle G\rangle \notin$ HAMILTON $\Rightarrow$ ogni tour T di G_c ha costo $\geq |V|+1 \Rightarrow$

$\Rightarrow A_{\Delta TSP}(\langle G_c, w\rangle, \frac{1}{2|V|})$ ritorna $\hat{T} : w(\hat{T}) \geq |V|+1$

Allora: $\langle G\rangle \in$ HAMILTON $\Leftrightarrow A_{\Delta TSP}(\langle G_c, w\rangle, \frac{1}{2|V|})$ ritorna $\hat{T}$ con $w(\hat{T}) \leq |V|$

$T_{A_{\Delta TSP}}(|\langle G_c, w\rangle|, \frac{1}{2|V|}) = poly(|\langle G_c, w\rangle|) = poly(|\langle G\rangle|)$

$\overset{!}{=} poly(2|V|) = poly(|\langle G\rangle|)$

$\Rightarrow poly(|\langle G\rangle|)$

$\Downarrow$

contraddizione con $P \neq NP$

EXERCISE 5 Provide a greedy-based, fast, 2-approximation algorithm for the optimization version of Subset_Sum

Recall: Instance is $\langle S, t \rangle$

$S = \langle x_1, x_2, \dots, x_n \rangle$, $x_i \le t$ w.l.o.g.

Determine

$$S^* = \arg\max \Big\{ \overset{\sum_{s \in S'} s}{\cancel{|S'|}} : S' \subseteq S, \sum_{s \in S'} s \le t \Big\}$$

Greedy approach based on a specific ordering of the elements of S: Keep selecting elements in the order while the sum stays $\le t$.

ordinare S in modo crecente: male

$S = \{x_1 = 2, x_2 = t-2\} \Rightarrow$ prendo 2, poi non posso prendere $t-2$

ordino S in modo decrescente

```
GREEDY(<S,t>):
  n ← |S|
  <x_1,...,x_n> ← DECREASING_SORT(S);          Θ(n log n)
  sum ← x_1;
  for i ← 2 to n do:
     if sum+x_i > t then return sum;
     else sum ← sum+x_i;
  return sum;
```

$sum \geq x_1$

se uso return fuori da ciclo $\Rightarrow sum = \sum_{s \in S} s \leq t \Rightarrow$ sol. ottima $\Rightarrow \rho = 1$

Supponiamo esista t.c. i che usa return:

$sum + x_i > t, \quad i \in [1, n] \Rightarrow x_i > t - sum, \quad sum \geq x_1 \Rightarrow$

$\Rightarrow sum \geq x_i > t - sum \Rightarrow 2 sum > t \Rightarrow sum > t/2$

$\rho = \frac{sum^*}{sum} < \frac{t}{t/2} = 2$

EXERCISE 6 Consider the following greedy algorithm returning an independent set of a graph $G=(V,E)$:

GREEDY_IS $(G=(V,E))$

$V' \leftarrow \emptyset$

$W \leftarrow V$

while $(W = \emptyset)$ do

* select arbitrary $v \in W$ *

$V' \leftarrow V' \cup \{v\}$

$W \leftarrow W - \{v\} - \{u \in W : (u,v) \in E\}$

return V'

Prove that:

1. V' is an IS of G

2. $\forall v \in V : \overbrace{(v \in V') \vee (\exists u \in V' : \{u,v\} \in E)}^{\text{def. dominating set}}$

   (V' is maximAL ($\neq$ maximUM) in the sense that $\forall v \in V - V'$: $V' \cup \{v\}$ is not independent set)

3. $\rho_{GIS} \leq \Delta = \max\{deg(v) : v \in V\}$

① contraddizione: suppongo V' non è IS $\Rightarrow \exists u, v \in V' : \{u, v \in E\} \Rightarrow$
$\Rightarrow$ poniamo che dg. nella prima u $\Rightarrow$ clean-up toglierebbe $v \Rightarrow$
$\Rightarrow$ contraddizione

② $v \in V' \Rightarrow$ ok
$v \notin V' \Rightarrow v$ eliminato in clean-up dopo $V' \leftarrow V' \in \{u\} \Rightarrow$
$\Rightarrow \{u, v\} \in E \Rightarrow$ ok

③ V^*: IS massimo, V': IS ritornato
prendo $v \in V^*$:
a) $v \in V'$
b) $v \notin V' \Rightarrow \exists u \in V' : \{u, v\} \in E$

$$V^* = \underbrace{(V^* \cap V')}_{X} \cup \underbrace{(V^* \setminus V')} = X \cup (V^* \setminus X) \Rightarrow |V^*| = x + (|V^* \setminus X|)$$

$|X| = x$

ogni $u \in V^* \setminus X$ deve essere connesso a nodo in $V' \setminus X \Rightarrow$
$\Rightarrow |V^* \setminus X| \leq \Delta(|V' \setminus X|)$

$$|V^*| = |X| + \Delta(|V' \setminus X|) \leq \Delta(|V'| + x - x) = \Delta|V'| \Rightarrow \rho = \frac{|V^*|}{|V'|} \leq \Delta$$

I.I: comunque prendo IS, posso partizionarlo in nodi che sono anche in massimale e nodi che non lo sono per DOM.SET, algoritmo di $(1+\Delta)$-approx.

possiamo anche fare DOMINATING.SET $\leq_p$ SC

$$(G=(V,E)) \rightsquigarrow (V,\mathcal{F}),\ \mathcal{F}=\{N_v=\{u:\{u,v\}\in E\}\cup\{v\}\}$$

$$\rho \leq H(\max_v |N_v|) = H(\Delta+1) = \Theta(\ln(\Delta+1))$$

EXERCISE 7 Given n jobs $J = \{1, 2, ..., n\}$ we wish to run them on m identical machines. Jobs have durations $t_1, t_2, ..., t_n$. We want to partition the jobs among the m machines so to minimize the maximum execution time of a single machine.

Formally: feasible solution:

schedule $\mathcal{P} = \{M_1, M_2, ..., M_m\}$:

$$M_i \cap M_j = \emptyset \qquad \bigcup_{k=1}^{m} M_k = \{1, ..., n\}$$

Execution time of M_i : $T_i = \sum_{j \in M_i} t_j$

Objective function:

minimize $T = \max \{T_i : 1 \le i \le m\}$

1. Prove that the decision version of the problem, LOAD BALANCING

$$\begin{cases} I: \langle \{t_1, t_2, ..., t_n\}, m, K \rangle \\ Q: \exists \text{ schedule } \mathcal{P} \text{ with } T \le K? \end{cases}$$

$\in$ NPH

(2) Let T^* be the cost of the optimal solution. Prove that

$$T^* \geq \max\left\{\left(\sum_{j=1}^{n} t_j\right)/m,\ \max_{1\leq j\leq n}\{t_j\}\right\}$$

(3) Give a 2-approximation algorithm for optimal LOAD BALANCING

---

(1) We prove

PARTITION $\leq_p$ LOAD BALANCING

$$\begin{cases} I: \langle S\rangle,\ S\subset \mathbb{N}^+,\ \text{finite} \\ Q: \exists\, S_1, S_2 \subset S,\ (S_1\cup S_2 = S)\wedge(S_1\cap S_2=\emptyset): \\ \qquad \sum_{s\in S_1} s = \sum_{s\in S_2} s\ ? \end{cases}$$

2 $\boxed{T^* \geq \max\left\{\left(\sum_{j=1}^{n} t_j\right)/m,\ \max_{1\leq j\leq n}\{t_j\}\right\}}$

(3) Greedy algorithm : allocate next job to the less loaded machine :