Versione riconoscimento e versione ottimizzazione

- Un problema di ottimizzazione richiede normalmente di trovare il minimo o il massimo di una funzione. Ad es.:
  risolvere un'istanza di LP;
  trovare il cammino più lungo da s a t in un grafo.

  Si parla di problemi in *Versione ottimizzazione, OV*.

- Un problema può anche richiedere solo risposta sì/no. Ad es.:
  stabilire se un'istanza di LP ha soluzione $\leq K$ (per K dato);
  stabilire se il cammino più lungo da s a t in un grafo
  ha valore $\geq K$ (per K dato).

  Si parla di problemi in *Versione riconoscimento, RV*.

- La teoria della complessità è stata sviluppata per i problemi in versione riconoscimento.

- Si dimostra però che

$$
\begin{gathered}
(\exists \text{ algoritmo polinomiale per un problema in RV}) \\
\Downarrow \\
(\exists \text{ algoritmo polinomiale per la corrispondente OV}),
\end{gathered}
$$

  e viceversa. Cioè:

- dal punto di vista della polinomialità o meno, le due versioni hanno la stessa difficoltà.

  (Intuitivamente: un problema in OV può essere risolto,
  utilizzando la *ricerca binaria*,
  mediante un numero polinomiale di interrogazioni successive)

Classi $\mathcal{P}$ ed $\mathcal{NP}$

- $\mathcal{P}$ = classe dei problemi in RV per cui $\exists$ algoritmo $\mathcal{P}$olinomiale.
  Es.: cammini minimi, minimo spanning tree;

  il simplesso ha *mediamente* tempo polin. anche se nel *caso peggiore* può esplorare tutte le basi $\rightarrow$ tempo $O\left(\dfrac{n!}{m!\,(n-m)!}\right)$,

  ma $\exists$ algoritmi polinomiali (meno efficienti) $\Longrightarrow$ LP $\in \mathcal{P}$.

- $\mathcal{NP}$ = classe dei problemi in RV tali che, se l'istanza ha risposta "sì", ciò può essere certificato in tempo polinomiale (es. esibendo un cammino da s a t di lunghezza $\geq K$), ossia c' è "speranza" di poter trovare un algoritmo polinomiale.

- In pratica, $\forall$ problema di ottimizzazione combinatoria $\in \mathcal{NP}$.
  $\mathcal{P} \subseteq \mathcal{NP}$.

- A$\in \mathcal{NP}$ è *trasformabile polinomialmente* in B$\in \mathcal{NP}$ se $\exists$ algoritmo polinomiale che, $\forall$ istanza di A, definisce un'istanza di B che ha sol. "sì" se e solo se l'istanza di A ha sol. "sì".

$$\Downarrow$$

- Se $\exists$ un algoritmo polinomiale per B, $\exists$ anche per A.

- Si dimostrato che praticamente tutti i problemi per i quali non si conoscono algoritmi polinomiali si trasformano polinomialmente uno nell'altro (*problemi $\mathcal{NP}$-completi*).

$$\Downarrow$$

- Se si trovasse un algoritmo polinomiale per uno qualunque di questi problemi, lo si avrebbe per tutti ($\Longleftrightarrow \mathcal{P} = \mathcal{NP}$).

$$\Downarrow$$

- È "improbabile" che $\exists$ tale algoritmo, e si ritiene che $\mathcal{P} \neq \mathcal{NP}$.