MA4701-1 - Optimización Combinatorial
Profesor: José Soto
Auxiliares: Nicolás Sanhueza - Christian von Borries

Auxiliar N°14

28 de Noviembre de 2013

Definición: Un problema de optimización P es o bien un problema de minimización o maximización. Cada instancia válida I del problema tiene un conjunto no vacío de soluciones factibles y cada solución factible un valor de la función objetivo. Llamaremos $\text{OPT}(I)$ al valor de una solución óptima de la instancia I.

Sea $\mathcal{A}$ un algoritmo que entrega siempre soluciones factibles (pero no necesariamente óptimas) y sea $\text{ALG}(I)$ el valor que entrega ese algoritmo en la instancia I. Diremos que $\mathcal{A}$ es una ρ-aproximación para P si para toda instancia I el algoritmo aproxima la solución óptima en un factor ρ. Es decir:

$$\text{OPT}(I) \leq \rho \cdot \text{ALG}(I) \quad \text{si } P \text{ era un problema de maximización}$$

$$\rho \cdot \text{OPT}(I) \geq \text{ALG}(I) \quad \text{si } P \text{ era un problema de minimización}$$

Nota: Por definición necesariamente $\rho \geq 1$, con igualdad ssi el algoritmo encuentra siempre un óptimo.

P1) Un hipergrafo k-uniforme es un par $G = (V, E)$, donde V es un conjunto finito de vértices y $E \subseteq \binom{V}{k}$ es una familia de subconjuntos de tamaño k de V llamados arcos. Una colección disjunta de arcos se llamará un matching. Sea $w : E \to \mathbb{R}^+$ una función de pesos de los arcos y considere el problema de buscar un matching de peso máximo. La meta de este problema es mostrar que el algoritmo glotón, que ordena los pesos arcos en forma decreciente y los agrega mientras que sigan formando un matching, es una k-aproximación del problema.

a) Sean $e_1, \ldots, e_l$ los arcos del matching engtregado por el algoritmo en el orden que son agregados. Sea M_i un matching de peso máximo entre los que contienen a $\{e_1, \ldots, e_i\}$. Muestre que, para todo $i = 1, \ldots, l$:

$$w(M_{i-1}) \leq w(M_i) + (k-1)w(e_i)$$

b) Concluya que el algoritmo es una k-aproximación del problema.

P2) **Árbol de Steiner** En el problema de Árbol de Steiner recibimos como instancia un un grafo $G = (V, E)$ no dirigido completo, $d : E \to \mathbb{R}^+$ una función de pesos de los arcos y $X \subseteq V$ un subconjunto de nodos. Buscamos el árbol $H = (U, T)$ con $X \subseteq U$ que sea de peso mínimo.

Restringiremos el problema al caso en el que la función d cumpla la desigualdad triangular, o sea:

$$\forall x, y, z \in V \quad d(x, z) \leq d(x, y) + d(y, z)$$

Considere el algoritmo que busca un árbol generador de peso mínimo para U e ignora el resto de V.

a) Muestre mediante un ejemplo que el algoritmo no necesariamente encuentra el óptimo.

b) Sea H solución óptima para el problema, $v \in H$ y $u_1, \ldots, u_l$ los nodos de U en el orden que los visitaría DFS(v). Entonces, para todo $i = 1, \ldots, l$

$$d(u_i, u_{i+1}) \leq d((u_i, u_{i+1})\text{-paseo en DFS}(v))$$

c) Muestre que el algoritmo es una 2-aproximación para el problema. ¿Es ajustada la desigualdad?

Definición: Un lenguaje A es **polinomialmente reducible** a un lenguaje B, escrito como $A \leq_p B$, si existe una función $f : \Sigma^* \to \Sigma^*$, tal que para todo $w \in \Sigma^*$

$$w \in A \iff f(w) \in B$$

y existe un algoritmo polinomial que en entrada w entregue $f(w)$.

P3) Sean A y B lenguajes tales que $A \leq_p B$ y suponga que tiene un algoritmo polinomial para B. Muestre que existe un algoritmo polinomial para A.

P4) Sean

$$L = \{\langle G, d, U, a\rangle \mid G \text{ grafo no dirigido}, d : E \to \mathbb{R}^+, U \subseteq V, \text{tal que existe un Árbol de Steiner } T \text{ con } w(T) \leq a\}$$

$$L' = \{w \in L \mid d \text{ cumple la desigualdad triangular}\}$$

a) Muestre que L y L' están en NP.

b) Muestre que $L \leq_p L'$.