Ma47a Optimización Combinatorial — **7 de Julio, 2006**

Examen

Prof. Cátedra: M. Kiwi — *Prof. Auxiliar: G. Sanchez*

TIEMPO: 4.0 HRS.

PROBLEMA 1: Pruebe el Teorema de Flujo-Máximo/Corte-Mínimo, i.e.

Teorema 1 *Si existe un (s,t)-flujo factible en la red $G=(V,E)$ con función de capacidad en los arcos $u : E \to \mathbb{R}_+$, entonces*

$$\max\{f_x(s) : x \text{ un } (s,t)\text{-flujo factible}\} = \min\{u(\delta(S)) : \delta(S) \text{ un } (s,t)\text{-corte}\} .$$

PROBLEMA 2: Sean $T_1, \ldots, T_n$ tareas y $M_1, \ldots, M_k$ máquinas. Cada tarea T_i requiere un tiempo de procesamiento p_i en alguna de las máquinas. Además es posible "preempción", i.e., T_i no necesariamente debe procesarse durante un intervalo seguido de largo $p_i \in \mathbb{N}$, ni siempre en la misma máquina. Sin embargo, debe procesarse en a lo más una máquina en cualquier instante de tiempo dado. La tarea T_i queda disponible para ser procesada en el instante $r_i \in \mathbb{Z}$ y debe ser completada a más tardar en el instante $d_i \in \mathbb{Z}$. Muestre como resolviendo un problema de flujo máximo se puede encontrar eficientemente cuándo, en qué máquina y por cuánto tiempo procesar las tareas.

Indicación: Ordenar los r_i's y d_i's en una secuencia de instantes $t_1 < t_2 < \ldots < t_{m+1}$, con $m+1 \leq 2n$, y considere los m intervalos de tiempo $I_j = (t_j, t_{j+1})$, $j \in \{1, \ldots, m\}$. Muestre como decidir cuanto del tiempo de procesamiento requerido por T_i se realizará durante el intervalo I_j, para $i \in \{1, \ldots, n\}$ y $j \in \{1, \ldots, m\}$.

PROBLEMA 3: Considere el Algoritmo 1.

Algorithm 1 Set Cover Glotón

Require: Ω conjunto finito, $\mathcal{S} \subseteq 2^{\Omega}$ colección de conjuntos y $c : \mathcal{S} \to \mathbb{Q}_+$ función de costos.
Ensure: $\bigcup_{S \in \mathcal{S}} S = \Omega$.

$C \leftarrow \emptyset$; $\mathcal{C} \leftarrow \emptyset$;
while $C \neq \Omega$ **do**
  $S' \leftarrow \operatorname{argmin}\left\{\frac{c(S)}{|S \setminus C|} : S \in \mathcal{S} \setminus \mathcal{C}\right\}$;
  for $\omega \in S' \setminus C$ **do**
    $\mathit{precio}(\omega) \leftarrow \frac{c(S')}{|S' \setminus C|}$;
  end for
  $C \leftarrow C \cup S'$; $\mathcal{C} \leftarrow \mathcal{C} \cup \{S'\}$;
end while
return $\mathcal{C}$;

Sea $\omega_1, \ldots, \omega_n$ un ordenamiento de Ω consistente con el orden en que fueron recubierto sus elementos.

(i).- (4.0 pts) Pruebe que $precio(\omega_k) \leq OPT/(n-k+1)$ para todo $k \in \{1, \ldots, n\}$.

(ii).- (2.0 pts) Concluya que el Algoritmo Set Cover Glotón es una $H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}$ aproximación para el problema de Set Cover.

Problema 4: Sea S un conjunto finito e $\mathcal{I}$ una familia de subconjuntos de S, llamados *independientes*. Decimos que $M = (S, \mathcal{I})$ es un *matroide* si

- (M0) $\emptyset \in \mathcal{I}$.
- (M1) Si $J' \subseteq J \in \mathcal{I}$, entonces $J' \in \mathcal{I}$.
- (M2) Para todo $A \subseteq S$, todo subconjunto independiente maximal (para la inclusión) de A tiene la misma cardinalidad.

(i).- (2.5 pts) Sea $G = (V, E)$ un grafo e

$$\mathcal{I} = \{J \subseteq E \,:\, G - J \text{ tiene el mismo número de componentes conexas que } G\} .$$

Pruebe que $M = (E, \mathcal{I})$ es un matroide.

(ii).- (3.5 pts) Considere el Algoritmo 2. Muestre que el Algoritmo de Kruskal (para la determina-

Algorithm 2 Algoritmo Glotón Genérico

Require: S conjunto finito, $\mathcal{I}$ colección de conjuntos independientes y $c : S \to \mathbb{Q}$ función de costos.

$J \leftarrow \emptyset$;
while $\exists s \notin J$ tal que $c(s) > 0$ y $J \cup \{s\} \in \mathcal{I}$ **do**
  $s' \leftarrow \text{argmax}\{s \,:\, c(s) > 0 \text{ y } J \cup \{s\} \in \mathcal{I}\}$;
  $C \leftarrow C \cup S'$; $J \leftarrow J \cup \{s'\}$;
end while
return J;

ción de un árbol generador de peso MÁXIMO) es una instancia del Algoritmo Glotón Genérico. (Especifique la función de costo $c(\cdot)$, el universo S, la colección de conjuntos independientes $\mathcal{I}$, y pruebe que $M = (S, \mathcal{I})$ es un matroide.)