**MA4701. Optimización Combinatorial. 2013.**
**Profesor:** José Soto

# Problemas Controlables (parte 2).

**Problema 7.** Una función $f\colon 2^S \to \mathbb{R}$ que satisface que para todo $X, Y \subseteq S$,

$$f(X\cup Y) + f(X\cap Y) \le f(X) + f(Y) \quad \text{se llama submodular,}$$
$$f(X\cup Y) + f(X\cap Y) = f(X) + f(Y) \quad \text{se llama modular,}$$
$$f(X\cup Y) + f(X\cap Y) \ge f(X) + f(Y) \quad \text{se llama supermodular.}$$

a) Demuestre que $f$ es submodular si y solo si satisface la siguiente condicion de retornos decrecientes

$$\forall X \subseteq Y, e \notin Y,\ f(Y+e) - f(Y) \le f(X+e) - f(X).$$

b) Demuestre que si $f$ es modular en $S$, entonces existen valores $(f_x)_{x\in S}$ tal que

$$\forall X \subseteq S,\ f(X) = f(\emptyset) + \sum_{x\in X} f_x.$$

c) Determine si son modulares, submodulares o supermodulares las siguientes funciones:

- La función $cc\colon 2^E \to \mathbb{R}$, donde $cc(X)$ = número de componentes conexas del grafo $(V, X)$ donde $G = (V, E)$ es un grafo dado.
- Dado un conjunto finito de eventos $S = \{A_1, A_2, \ldots, A_k\}$ en un espacio de probabilidad cualquiera, estudie la función $f\colon 2^{[k]} \to \mathbb{R}$ dada por $f(X) = \Pr(\bigwedge_{i\in X} A_i)$.
- Dada una función $g\colon \mathbb{R} \to \mathbb{R}$ convexa y una función $h\colon 2^V \to \mathbb{R}$ modular, analice la funcion $f\colon 2^V \to E$ dada por $f(X) = g(h(X))$.

d) Sea $f\colon 2^{[n]} \to \mathbb{R}$ una función sobre los subconjuntos de $[n]$. Definimos la extensión multilinear $\hat{f}$ de $f$ como $\hat{f}\colon [0,1]^n \to \mathbb{R}$, $\hat{f}(x) = \mathbb{E}[f(X)]$ donde $X \subseteq [n]$ es el conjunto aleatorio dado por: $i \in X$ con probabilidad $x_i$ de manera independiente.

Demuestre que $f$ es submodular si y solo si $\frac{\partial^2 \hat{f}}{\partial x_i \partial x_j} \le 0$, para todo $i, j \in [n]$.

**Nota: En general, $\hat{f}$ no es cóncava ni convexa. ¡Sin embargo, tiene todas sus segundas derivadas parciales principales no positivas!**

**Problema 8.** [Programación Dinámica]

a) Dado un vector $v \in \mathbb{R}^n$, diseñe un programa dinámico que determine el subvector $v[i,j] = (v_i, v_{i+1}, \ldots, v_j)$ de mayor suma. ¿Cuál es la complejidad de su algoritmo? (Su algoritmo debe ser más eficiente que el algoritmo trivial que calcula la suma de todos los $v[i,j]$ de manera independiente que toma tiempo $O(n^3)$).

b) Dada una secuencia $S = v_1, v_2, \ldots, v_n$ de enteros en el rango $\{0, 1, \ldots, k\}$ diseñe un programa dinámico que particione $S$ en dos subconjuntos $S_1$ y $S_2$ con sumas parciales $s_1 = \sum_{v\in S_1} v$ y $s_2 = \sum_{v\in S_2} v$ tal que $|s_1 - s_2|$ sea mínimo. Su algoritmo debe ser polinomial en $n$ y lineal en $k$.
**Ayuda**: Inspírese en la secuencia de preguntas: ¿Posee $S$ un subconjunto de suma $j$?

**Problema 9.** Sea $S = v_1, v_2, \ldots, v_n$ una secuencia finita de $n$ enteros. Una subsecuencia de $S$ es una secuencia $v_{i_1}, \ldots, v_{i_m}$ con $i_1 < \cdots < i_m$. Esta secuencia se dice creciente si además $v_{i_1} < \cdots < v_{i_m}$.

a1) Formule el problema de calcular la subsecuencia creciente más larga de $S$ como un problema de camino de largo máximo en un digrafo acíclico. Usando esto, diseñe un algoritmo que calcule el largo de la subsecuencia creciente más larga de $S$ en tiempo $O(n^2)$.

a2) Diseñe un algoritmo para encontrar la subsecuencia creciente de menor suma de $S$ (note que la suma puede ser negativa).

Deseamos resolver un sistema $S$ de ecuaciones con $n$ variables $x_i$, y $m$ ecuaciones del tipo $x_i - x_j \leq c_{i,j}$, donde $c_{i,j} \in \mathbb{R}$ es constante. Construya el digrafo $G = (V, E)$ con $n$ nodos $v_i$ asociados a las variables $x_i$ y $m$ arcos $(i, j)$ asociados a la ecuacion de constante $c_{i,j}$. Asigne además a cada arco $(i, j)$, largo $c(i, j)$.

b1) Demuestre que $S$ no tiene solución si y solo si $G$ tiene un ciclo de costo negativo.

b2) Diseñe un algoritmo para encontrar una solución de $S$ si esta existe, y que además reporte si no hay solución. **Ayuda:** Agregue un nodo extra $v_0$ a $G$ que denote el valor 0, y arcos desde $v_0$ a los otros nodos con pesos adecuados. Interprete el problema como uno de caminos de largo mínimo.

**Problema 10.** [Algoritmo de Strassen para multiplicar matrices.] Dadas dos matrices $A \in \mathbb{Z}^{n\times n}$ y $B \in \mathbb{Z}^{n\times n}$, calcular la matriz $AB$ se puede hacer trivialmente en $O(n^3)$ multiplicaciones y sumas. En esta pregunta mostraremos una técnica para calcular el producto de $A$ y $B$, en el caso $a = b = c =: n = 2^t$ usando una cantidad subcúbica de multiplicaciones y sumas (comentario: Si los enteros involucrados están acotados por una cantidad fija, entonces este algoritmo es verdaderamente fuertemente polinomial y corre en tiempo $o(n^3)$). Sea $t \geq 0$, $n = 2^t$ y $A, B \in \mathbb{Z}^{n\times n}$, note que podemos escribir

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix}, B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix},$$

donde $A_{i,j}, B_{i,j} \in \mathbb{Z}^{n/2\times n/2}$. Definimos las siguientes matrices auxiliares

$$\begin{aligned}
Q_1 &= (A_{1,1} + A_{2,2})(B_{1,1} + B_{2,2}) \\
Q_2 &= (A_{2,1} + A_{2,2})B_{1,1} \\
Q_3 &= A_{1,1}(B_{1,2} - B_{2,2}) \\
Q_4 &= A_{2,2}(-B_{1,1} + B_{2,1}) \\
Q_5 &= (A_{1,1} + A_{1,2})B_{2,2} \\
Q_6 &= (-A_{1,1} + A_{2,1})(B_{1,1} + B_{1,2}) \\
Q_7 &= (A_{1,2} - A_{2,2})(B_{2,1} + B_{2,2}).
\end{aligned}$$

a) Demuestre que $AB = \begin{pmatrix} Q_1 + Q_4 - Q_5 + Q_7 & Q_3 + Q_5 \\ Q_2 + Q_4 & Q_1 + Q_3 - Q_2 + Q_6 \end{pmatrix}$.

b) Llame $f(t)$ al número de multiplicaciones escalares necesarias para multiplicar dos matrices de dimensiones $2^t \times 2^t$ mediante la fórmula calculada anteriormente. Calcule $f(0)$ y encuentre una recursión para $f$ de la forma $f(t) = 7f(t-1) + dt^2$.

c) Pruebe que $f(t) = O(7^t)$ y deduzca que el algoritmo resuelve el problema usando $O(n^{\log_2 7}) \approx O(n^{2{,}807})$ operaciones.

d) Suponga que $G$ es un grafo simple. Es fácil crear un algoritmo que determine si $G$ contiene un triángulo en $O(n^3)$ o incluso en $O(nm)$ operaciones (inténtelo). Usando las partes anteriores, diseñe un algoritmo para determinar si $G$ contiene un triángulo en tiempo $O(n^{\log_2 7})$. (**Ayuda:** Suponga primero que $G$ tiene exactamente $2^t$ vértices).

**Problema 11.**

a) Sea $G = (V, E)$ un grafo simple **no** dirigido y no acíclico, y $\ell\colon E \to \mathbb{R}_+$ una funcion de largos no negativos. Diseñe un algoritmo para encontrar el ciclo sde $G$ de largo minimo. ¿Cuán rápido puede lograr que sea su algoritmo? Tenga cuidado: Simplemente reemplazar aristas por arcos dirigidos genera ciclos dirigidos de 2 arcos que NO corresponden a aristas en el grafo original.

b) La cintura de un grafo $G$ simple y no dirigido se define como el mínimo número de vértices en un ciclo de $G$ (o infinito, si $G$ es acíclico). Diseñe un algoritmo para encontrar la cintura de $G$. Su algoritmo debe ser más rápido que el usado en la parte anterior.

**Problema 12.**
Sea $G = (V, E)$ un grafo simple. Considere el par $(V, \mathcal{I})$ donde $X \in \mathcal{I}$ si y solo si existe un matching $M$ que cubre todos los elementos de $X$.

a) Demuestre que el par $(V, \mathcal{I})$ es una matroide.

b) Suponga ahora que $G$ es bipartito con bipartición $V = A \dot\cup B$. Diseñe un algoritmo que dado $X \subseteq A$ determine el rango de $X$ en $(V, \mathcal{I})$.

c) Todavía en el caso bipartito, diseñe un algoritmo que dado $X \subseteq A$, encuentre $\mathrm{span}(X)$.

**Problema 13.**

a) Demuestre el Teorema de matrimonio de Hall directamente usando el Teorema de König.

b) Sea $G = (V, E)$ un grafo con bipartición $A \dot\cup B$. Para todo $X \subseteq A$, defina el defecto de $X$ como $\mathrm{def}(X) = |X| - \Gamma(X)$ con $\Gamma(X) = |N(X)|$. Muestre que el defecto 'def' es supermodular.

c) Sea $d^* = \text{máx}_{X \subseteq A} \mathrm{def}(X)$. Generalice el teorema de Hall mostrando que el tamaño máximo de un matching en $G$ es igual a $|A| - d^*$.

**Problema 14.** Sea $G = (V, E)$ un grafo dirigido simple y completo (es decir, $E = V \times V$) y $w\colon E \to \mathbb{R}$ una función de peso cualquiera. Un cubrimiento (exacto) por ciclos de $G$ es una colección de ciclos dirigidos $\mathcal{C}$ de $G$, tal que cada vértice $v \in V$ aparece exactamente un ciclo $C \in \mathcal{C}$. Definimos además el peso de $\mathcal{C}$ como

$$\sum_{C \in \mathcal{C}} w(C) = \sum_{C \in \mathcal{C}} \sum_{e \in E(C)} w(e).$$

Diseñe un algoritmo polinomial que encuentre un cubrimiento por ciclos de $G$ de peso mínimo.
**Indicación:** Reduzca el problema anterior a un problema de asignación en un grafo bipartito.

**Problema 15.** Sea $G = (V, E)$ un grafo bipartito y considere los polítopos

$$P(G) = \left\{x \in \mathbb{R}^E : x \geq 0 \text{ y } \sum_{e \in \delta(v)} x_e \leq 1 \,\forall v \in V\right\} \text{ y } P_{\text{matching}}(G) = \mathrm{conv}(\{\chi_M : M \text{ matching en } G\}).$$

Demuestre que $P(G) = P_{\text{matching}}$.
**Indicación:** Para probar la inclusión no trivial, considere la siguiente construcción auxiliar. Sean $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ dos copias disjuntas de $G$ con $V_i = \{v_i : v \in V\}$, $i \in \{1, 2\}$. Defina un nuevo grafo $H$ con $V(H) = V_1 \cup V_2$ y $E(H) = E_1 \cup E_2 \cup \{(v_1, v_2) : v \in V\}$. Para probar la propiedad, encuentre una relación entre los matchings de $G$ y los matchings perfectos de $H$; y una relación similar entre los vectores de $P(G)$ y los vectores de $P_{\text{match. perfecto}}(H)$, cuya caracterización se vió en clases.