



Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática
MA47A: Optimización Combinatorial
Profesor: Roberto Cominetti
Auxiliares: Raul Aliaga Diaz, Cristóbal Guzmán

Guia 02

P1. Flujo en grafos

Consideremos el problema de flujo de costo mínimo en un grafo capacitado $G = (V, E)$,

$$(FCM) \quad \min_{f \in \mathbb{R}^{|E|}} \{c^t f : \text{div}(f) = b, l \leq f \leq u\},$$

donde $b \in \mathbb{R}^{|V|}$ representa la oferta en los nodos y $l, u \in \mathbb{R}^{|E|}$ las capacidades mínimas y máximas en los arcos. Supongamos que los vectores b, l y u tienen coordenadas enteras.

- Utilice el algoritmo de flujo de costo mínimo visto en clases para probar que (FCM) tiene al menos una solución óptima f^* con coordenadas enteras.
- Sea A la matriz de incidencia

$$A_{ie} = \begin{cases} 1 & \text{si } i = I(e), \\ -1 & \text{si } i = T(e), \\ 0 & \text{si no,} \end{cases}$$

de modo tal que $\text{div}(f) = Af$.

- Probar que cualquier submatriz cuadrada B de A satisface $\det(B) \in \{-1, 0, 1\}$. Una matriz con esta propiedad se dice *totalmente unimodular*.

Sugerencias:

- Razone por inducción sobre la dimensión de B .
- Observar que la suma de las filas de A es el vector 0.

- Consideremos el caso particular $l = 0, u = \infty$. Utilice la parte (a) para probar que todo punto extremo del polítopo

$$F := \{f \in \mathbb{R}^{|E|} : \text{div}(f) = b, l \leq f \leq u\}$$

tiene coordenadas enteras.

Sugerencia: recuerde la correspondencia entre los puntos extremos de un polítopo y las soluciones básicas factibles. Notar que $\text{rg}(A) = |V| - 1$.

P2. Un cubrimiento por arcos de un grafo $G = (V, E)$ es un conjunto $C \subset E$ tal que

$$V = \bigcup_{(u,v) \in C} \{u, v\}.$$

Suponiendo que G no tiene puntos aislados,

- (a) pruebe que $|C^*| = |V| - |M^*|$ donde C^* es un cubrimiento por arcos de cardinal mínimo, y M^* es un matching de cardinal máximo.
- (b) proponga un algoritmo eficiente para encontrar un cubrimiento por arcos de cardinal mínimo.

P3. Sea $G = (V, E)$ un grafo completo no-dirigido. Para cada arco $(i, j) \in E$ sea d_{ij} su longitud y B su capacidad (note que B es el mismo para todos los arcos y que $d_{ij} = d_{ji}$). Además, para cada $i \in V$ existe un flujo $A_i \leq B$ hacia un nodo fijo $s \in V$ que llamamos nodo central. Consideremos el problema de determinar un árbol de peso mínimo capaz de conducir el flujo desde todos los nodos hacia el nodo central, sin sobrepasar la capacidad de los nodos. Demuestre que en el caso que todos los A_i son 0 o 1 y $B = 1$, este problema se puede formular como un flujo de costo mínimo.

P4. *Programación de tareas*

Sea $\mathcal{A} = \{a_1, \dots, a_n\}$ un conjunto de tareas que deben ser ejecutadas en 2 máquinas. Cada tarea requiere de una unidad de tiempo para ser realizada, y es necesario satisfacer un conjunto de restricciones de precedencia representadas por medio de un grafo acíclico $P = (\mathcal{A}, \mathcal{P})$ en el que $(a, b) \in \mathcal{P}$ ssi la tarea a debe ejecutarse antes que b .

Una programación es una función $S : \mathcal{A} \rightarrow \{1, \dots, T\}$ que asigna a cada tarea el instante de ejecución, de modo tal que

- (1) $\forall i = 1, \dots, T \quad |\{a \in \mathcal{A} : S(a) = i\}| \leq 2,$
- (2) $(a, b) \in \mathcal{P} \Rightarrow S(a) < S(b).$

Deseamos encontrar una programación que minimice el tiempo total de ejecución T .

(a) Sea $G_P = (\mathcal{A}, E)$ donde $(a, b) \in E$ ssi no existe en P ningún camino de a a b ni tampoco de b a a (es decir, no hay ninguna relación de precedencia entre las tareas a y b). Sea M un matching máximo en G_P . Probar que toda programación satisface $T \geq |\mathcal{A}| - |M|$.

(b) Probar que el T óptimo es $T = |\mathcal{A}| - |M|$.

INDICACIÓN: Considerar un matching máximo $M = \{(a_1, b_1), \dots, (a_k, b_k)\}$. Probar que si se tiene una incompatibilidad del tipo

- a) a_i debe realizarse antes que a_j

b) b_j debe realizarse antes que b_i

entonces se puede intercambiar b_i con b_j , es decir podemos reemplazar $(a_i, b_i), (a_j, b_j)$ por $(a_i, b_j), (a_j, b_i)$, obteniéndose un nuevo matching máximo. Concluir.

P5. *Flujo de costo mínimo*

Un restaurante necesita una cantidad conocida de manteles $r_1, \dots, r_n$ para los próximos n días. Al final de cada día se debe decidir cuantos manteles sucios son enviados a la lavandería rápida (m días, costo f por mantel), cuantos a la lavandería lenta ($n > m$ días, costo $s < f$ por mantel), y cuantos son guardados sucios. La demanda de manteles es satisfecha a partir de los manteles limpios provenientes de cada lavandería, y en caso que falten es posible comprar manteles en el mercado M a un precio p . Inicialmente el restaurante no tiene manteles.

(a) Formular el problema anterior como un flujo de costo mínimo (considerar n nodos oferta, n nodos demanda,...)

(b) Resolver el caso $n = 3, r_1 = 3, r_2 = 2, r_3 = 4, m = 1, n = 2, f = 6, s = 3, p = 10$ mediante cualquier método (intuitivamente si quiere), pero probando la correctitud de su respuesta.