MA-47A Optimización Combinatorial	**8 de Julio, 2003**

Examen

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

TIEMPO: 4.0 HRS.

PROBLEMA 1:

(i).- (1.5 pts) Considere el siguiente problema:

MAX-CUT

Entrada:	$G = (V, E)$ grafo y $\omega : E \to \mathbb{R}$ función de peso
Solución:	$V' \subseteq V$
Costo:	$\sum_{e \in \delta(V')} \omega(e)$
Objetivo:	MAX

Se sabe que MAX-CUT es NP-duro (no lo pruebe). Proponga un algoritmo a tiempo polinomial de 2–aproximación para MAX-CUT. Pruebe que su algoritmo efectivamente entrega la aproximación deseada y explicite el tiempo que toma.

(ii).- (1.5 pts) Pruebe el teorema del flujo–máximo corte–mínimo para redes.

(iii).- (0.6 pts) Señale el error en el siguiente argumento (justifique su respuesta):

> A una instancia de MAX-CUT dada por G y ω le podemos asociar un problema de corte–mínimo sobre el mismo grafo G pero con pesos $-\omega(e)$ para cada arco $e \in E$. Luego, MAX-CUT puede reducirse eficientemente al problema de la determinación de un flujo máximo, que se puede resolver eficientemente. En resumen, MAX-CUT puede resolverse en tiempo polinomial.

(iv).- (1.2 pts) Sea $G = (V, E)$ un grafo (no–dirigido) con función de capacidad $u : E \to \mathbb{R}_+$. Suponga que hay k tipos de mercancías y una demanda d_i del nodo s_i al nodo t_i, $s_i \neq t_i$, por cada tipo i de mercancía. El problema del *flujo concurrente máximo* (MAX-FC) consiste en determinar el valor, digamos α^*, que maximiza α tal que para todo $i \in \{1, \ldots, k\}$ es factible enviar αd_i flujo de s_i a t_i, sujeto a que:

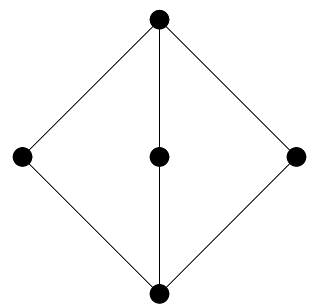
> (1) para toda mercancía i, y todo nodo en $v \in V \setminus \{s_i, t_i\}$, se conserve el flujo de la mercancía i en el nodo v,
>
> (2) en todo arco $e \in E$, la suma de los flujos (de las distintas mercancías) que pasan por e no excede $u(e)$.

Formalice el problema del MAX-FC y concluya que puede resolverse en tiempo polinomial.

(v).- (1.2 pts) Considere la notación y conceptos introducidos en el punto anterior y pruebe que

$$\alpha^* \leq \beta^* = \min\left\{ \frac{u(\delta(S))}{\sum_{i\in I(S)} d_i} : S \subseteq V, I(S) = \{i \,:\, |S \cap \{s_i, t_i\}| = 1\} \neq \emptyset \right\}.$$

Muestre, con un contraejemplo, que la igualdad no necesariamente se cumple. Para ello, considere 4 tipos de mercancías, demandas y capacidades iguales a 1 y la red de más abajo. Justifique.



Nota: Al problema de determinar β^* se le conoce por problema del multi–corte de razón mínima (y es NP-duro – no lo pruebe).

PROBLEMA 2: En lo que sigue considere el siguiente problema:

SET-COVER
Entrada: U conjunto finito, $\mathcal{S} \subseteq 2^U$, $\omega : \mathcal{S} \to \mathbb{R}_+$ función de peso
Solución: $\mathcal{C}$ tal que $U \subseteq \cup_{C\in\mathcal{C}} C$
Costo: $\sum_{C\in\mathcal{C}} \omega(C)$
Objetivo: MIN

(i).- Reduzca VERTEX-COVER al problema de decisión asociado a SET-COVER. Concluya que el referido problema de decisión es NP–completo.

(ii).- Considere los siguientes PL primal-dual

$$\begin{array}{llll} \text{(P)} & \text{Min} & c^T x \\ & \text{s.a.} & Ax \geq b \\ & & x \geq 0 \end{array} \qquad \begin{array}{llll} \text{(D)} & \text{Max} & b^T y \\ & \text{s.a.} & A^T y \leq c \\ & & y \geq 0 \end{array}$$

Sean x e y soluciones primal y dual factibles respectivamente que satisfacen, para $\alpha, \beta \geq 1$.

Holgura complementaria primal: $x_j = 0$ o $c_j/\alpha \leq \sum_i a_{i,j} y_i \leq c_j$.

Holgura complementaria dual: $y_i = 0$ o $b_i \leq \sum_j a_{i,j} x_j \leq \beta b_i$.

Pruebe que $c^T x \leq \alpha\beta b^T y$.

(iii).- Considere el siguiente esquema de aproximación primal–dual:

Aprox-Set-Cover
$x = (x_S : S \in \mathcal{S}) \longleftarrow 0$;
$y = (y_u : u \in U) \longleftarrow 0$;
$V \longleftarrow \emptyset$;
Mientras $\exists u \in U \setminus V$
Incrementar y_u hasta que $\sum_{u \in S} y_u = \omega(S)$ para algún $S \in \mathcal{S}$;
Hacer x igual al vector indicatriz de los S tales que $\sum_{u \in S} y_u = \omega(S)$;
$V \longleftarrow \bigcup_{S : x_S \neq 0} S$;
Return $\mathcal{C} \subseteq \mathcal{S}$ representada por x;

Pruebe que Aprox-Set-Cover es un algoritmo de f–aproximación para Set-Cover donde $f = \max\{f_u : u \in U\}$ y f_u denota la cantidad de conjuntos $C \in \mathcal{C}$ a los que pertenece u.