MA-47A Optimización Combinatorial	**15 de Marzo, 2000**

Pauta Control No. 1

Prof. Cátedra: M. Kiwi — *Prof. Auxiliar: J. R. Correa*

PROBLEMA 1:

(i).- Por inducción en $n = |V|$. Si $n = 1$, no hay nada que probar. Supongamos que se tiene para n. Sea $G = (V, E)$ un grafo no dirigido conexo tal que $|V| = n + 1$ y sea además $v \in V$ un nodo cualquiera. Denotemos por $V_1, \ldots, V_k$ las componentes conexas de $G|_{V \setminus \{v\}}$. En particular, $(V_i \times V_j) \cap E = \emptyset$ si $i \neq j$. Observar que

$$|V_i| \;\leq\; \sum_{i=1}^{k} |V_i| \;=\; |V| - 1 \;=\; n\,.$$

Luego, por hipótesis inductiva $|E(G(V_i))| \geq |V_i| - 1$. Por otro lado, como G es conexo debe existir en G al menos un arco que incida en v y en un nodo en V_i, para todo $i = 1, \ldots, k$. Sigue que,

$$|E(G)| \;\geq\; k + \sum_{i=1}^{k} |E(G_i)| \;\geq\; k + \sum_{i=1}^{k} (|V_i| - 1) \;\geq\; \sum_{i=1}^{k} |V_i| \;=\; |V| - 1\,.$$

(ii).- Si removemos el arco $e \in E(T)$ de T se obtienen dos componentes conexas. Si R denota el conjunto de nodos de una de estas componentes, entonces la otra componente conexa tiene por nodos $V(G) \setminus R$. Como los extremos de e pertenecen a componentes conexas distintas, sigue que

$$w(e) \quad \geq \quad \min \{ w(f) : f = (u,v) \in E, u \in R, v \in V(G) \setminus R \}\,.$$

Supongamos que la igualdad no se tiene. Entonces, existe $f = (u, v) \in E$ tal que $u \in R$, $v \in V(G) \setminus R$, y $w(e) > w(f)$. Más aun, si a T le quitamos e y agregamos f, entonces obtenemos un árbol T' que genera G y cuyo peso es $w(T') = w(T) - w(e) + w(f) < w(T)$ lo que contradice la minimalidad de T.

(iii).- Supongamos que G es bipartito y que $c = (u_0, \ldots, u_{l-1})$ es un ciclo en G. Queremos probar que el largo de c es par, i.e., que l es par. Observar que $l \neq 1$ porque un grafo bipartito no tiene bucles. Sin pérdida de generalidad supongamos que u_0 está en L. Como $(u_0, u_1) \in E$ y $E \subseteq (L \times R) \cup (R \times L)$ debe tenerse que $u_1 \in R$. El mismo argumento muestra que si $l > 2$ entonces u_2 está en L, y en general que u_{2i} está en L mientras que u_{2i+1} está en R. Como asumimos que u_0 está en L y c es un ciclo, debe tenerse que u_{l-1} está en R. Por lo tanto, l debe ser par.

Supongamos ahora que G sólo posee ciclos de largo par. Asumiremos que G es conexo pues de lo contrario basta repetir el argumento en cada componente conexa y observar que una colección de grafos bipartitos sigue siendo un grafo bipartito. Sea entonces s un nodo cualquiera en G, asignemos peso 1 a cada arco de G, y denotemos por $\delta(s, v)$ la distancia de s a $v \in V$. Como G es conexo y no dirigido $\delta(s, v) < +\infty$ para todo $v \in V$. Definamos

$$L \;=\; \{ v \in V \,:\, \delta(s,v) \text{ es par} \}\,, \qquad \text{y} \qquad R \;=\; \{ v \in V \,:\, \delta(s,v) \text{ es impar} \}\,.$$

Observemos que no puede haber un arco (u, v) con ambos extremos en L pues de lo contrario habría un ciclo en G (de s a u, de u a v, y de v a s) de largo $\delta(s,u) + 1 + \delta(s,v)$, i.e., de largo impar. Análogamente, no puede haber un arco con ambos extremos en R. Hemos establecido que $E \subseteq (L \times R) \cup (R \times L)$, luego G es bipartito.

(iv).- En la Fig. 1 se muestra una modificación del algoritmo de BELLMAN-FORD que, como se desprende de la discusión que sigue, retorna un nodo v_0 de G en un ciclo de peso negativo. El resto de los nodos en dicho ciclo son $v_1, \ldots, v_{l-1}$ donde $\pi(v_i) = v_{i+1 \bmod l}$ si $i \in \{0, \ldots, l-1\}$.

```
ENTRADA:      G = (V,E), s ∈ V, y w:E → ℝ.
CÓDIGO:   1.  for all v ∈ V do
          2.        d(v) ← ∞, π(v) ← NIL
          3.  d(s) ← 0
          4.  for i = 1 to |V| − 1 do
          5.        for all (u,v) ∈ E do
          6.             if d(v) > d(u) + w(u,v)
          7.                  then d(v) ← d(u) + w(u,v); π(v) ← u
          8.  for all v ∈ V do
          9.        Visit(v) ← FALSE
         10.  for all (u,v) ∈ E do
         11.        if d(v) > d(u) + w(u,v)
         12.             then π(v) ← u; break
         13.  while Visit(v) ≠ TRUE do
         14.        Visit(v) = TRUE; v ← π(v)
         15.  return v
```

Figure 1: BELLMAN-FORD modificado.

De la corrección del algoritmo de BELLMAN-FORD se desprende que si en el paso 11 se tiene que $d(v) > d(u) + w(u,v)$ entonces existe un ciclo de peso negativo en G. Además, si $\delta_k(s,v)$ es el peso mínimo de un camino que va de s a v y que ocupa exactamente k arcos, entonces, del análisis del algoritmo de BELLMAN-FORD se desprende que en el paso 12,

$$d(u) + w(u,v) \quad < \quad \min_{0 \leq k < |V|} \delta_k(s,v)\,,$$

y que existe un camino $p_{s,u}$ en G que va de s a u tal que $d(u) = w(p_{s,u})$. Sigue que el camino $p_{s,v}$ que va de s a v primero a lo largo de $p_{s,u}$ y luego tomando el arco (u,v) es tal que

$$w(p_{s,v}) \;=\; d(u) + w(u,v) \;<\; \min_{0 \leq k < |V|} \delta_k(s,v)\,.$$

Sigue que $p_{s,v}$ ocupa al menos $|V|$ arcos y que en dicho camino debe haber un ciclo c de peso negativo.

Partiendo de $v_0 = v$ y recorriendo los nodos de G de acuerdo a lo indicado por π, i.e., pasando de v_i a $v_{i+1} = \pi(v_i)$, eventualmente caemos en un ciclo c. En efecto, la única forma posible de no caer en un ciclo es llegar a s. Pero esto es imposible porque implicaría que la secuencia de nodos $v_0 = v, v_1, \ldots, s$ recorrida en sentido contrario es un camino que ocupa menos de $|V|$ arcos y cuyo peso es $w(p_{s,v}) < \min_{0 \leq k < |V|} \delta_k(s,v)$. Falta ver que c tiene peso negativo. Para ello supongamos que $c = (v_0, \ldots, v_{l-1})$, $\pi(v_i) = v_{i+1 \bmod l}$, y asumamos, sin pérdida de generalidad, que el último nodo del ciclo para el cual $\pi(v)$ se actualizó fue v_0. Sean $d'(v_0)$ el penúltimo valor que toma $d(v_0)$. Luego, para concluir que $w(c) < 0$ basta observar que

$$\begin{aligned} d'(v_0) &> d(v_1) + w(v_0, v_1) \\ &= d(v_2) + w(v_0, v_1) + w(v_1, v_2) \\ &= \dots \\ &= d'(v_0) + \sum_{i=0}^{l-1} w(v_i, v_{i+1 \bmod l}) \,. \end{aligned}$$

El número de operaciones adicionales que realiza la modificación propuesta de BELLMAN-FORD es $O(|V|)$ para implementar los pasos 8 y 9, $O(1)$ para implementar el paso 12, y $O(|V|)$ para implementar los pasos 13 y 14. Es decir, en BELLMAN-FORD se requiere un tiempo adicional de $O(|V|)$ para encontrar un ciclo de peso negativo.

PROBLEMA 2:

(i).- Claramente $\delta(s,v) \leq \delta_k(s,v)$, luego $\delta(s,v) \leq \min_{0\leq k\leq n-1} \delta_k(s,v)$.

Por otro lado, si $\mu^* = 0$, entonces $\mu(c) \geq 0$ para todo ciclo c. Sea p un camino de peso mínimo de s a v. Supongamos, para efectos de obtener una contradicción que p ocupa $k \geq n$ arcos. Necesariamente, p debe pasar por algún nodo al menos dos veces. Eliminando los arcos del ciclo entre dos retornos sucesivos a un mismo nodo y puesto que todo ciclo tiene peso no negativo obtenemos un camino de peso a lo más $w(p)$ con menos arcos. Repitiendo el argumento, hasta que el nuevo camino ocupe $k < n$ arcos, vemos que existe un camino de peso mínimo de s a v que pasa por a lo más $n-1$ arcos, i.e., $\delta(s,v) = \delta_k(s,v)$ para algún $k \in \{0,\dots,n-1\}$.

(ii).- Si $\delta_n(s,v) = +\infty$, entonces la desigualdad es obvia. Supongamos entonces que $\delta_n(s,v) < +\infty$ y sea p camino de s a v que pasa por n arcos y tal que $w(p) = \delta_n(s,v)$. Nuevamente, tenemos que p debe pasar por algún nodo más de una vez. Eliminando los arcos del ciclo entre dos retornos sucesivos a un mismo nodo y puesto que todo ciclo tiene peso no negativo (ya que $\mu^* = 0$) obtenemos un camino p' que pasa por $l \in \{0,\dots,n-1\}$ arcos y tal que $\delta_l(s,v) \leq w(p') \leq w(p) = \delta_n(s,v)$. Sigue que

$$\max_{0\leq k\leq n-1} \frac{\delta_n(s,v) - \delta_k(s,v)}{n-k} \;\geq\; \frac{\delta_n(s,v) - \delta_l(s,v)}{n-l} \;\geq\; 0\,.$$

(iii).- Sean $p_{u,v}$ y $p_{v,u}$ los caminos que van de u a v y de v a u a través del ciclo c. Claramente, $w(p_{u,v}) = \delta$. Dado que $\mu(c) = 0$ implica que $w(p_{u,v}) + w(p_{v,u}) = 0$ tenemos que $w(p_{u,v}) = -\delta$. Como para cualquier x e y en V se tiene que $\delta(s,x) \leq \delta(s,y) + \delta(y,x)$, sigue que $\delta(s,v) \leq \delta(s,u) + \delta(u,v) \leq \delta(s,u) + w(p_{u,v}) = \delta(s,u) + \delta$ y que $\delta(s,u) \leq \delta(s,v) + \delta(v,u) \leq \delta(s,v) + w(p_{v,u}) = \delta(s,v) - \delta$. En resumen, $\delta(s,v) = \delta(s,u) + \delta$.

(iv).- Si $\mu^* = 0$, entonces existe un ciclo c tal que $\mu(c) = 0$. Probemos primero la indicación. Para ello consideremos un nodo v en c. Sea $p_{s,v}$ un camino de peso mínimo que va de s a v (dicho camino existe pues todo nodo en G es alcanzable desde s). Observar que el camino $p'_{s,v}$ que primero recorre $p_{s,v}$ y luego c es un camino que va de s a v cuyo peso es $w(p_{s,v}) = w(p_{s,v}) + w(c) = \delta(s,v)$, i.e., también es un camino de peso mínimo de s a v. Como el camino que primero recorre $p_{s,v}$ y luego pasa al próximo arco del ciclo es un sub–camino del camino óptimo $p'_{s,v}$, por sub–optimalidad de los caminos óptimos, también es un camino de peso mínimo.

Tomemos ahora un nodo u en c. Por (i) sabemos que debe haber un camino p de s a u de peso mínimo que consta de $l \leq n-1$ arcos. Sea v el nodo en c que se alcanza desde u recorriendo a lo largo del ciclo $n-l$ arcos. Por la indicación, dicho camino debe ser de peso mínimo, luego $\delta_n(s,v) = \delta(s,v)$. De (i) concluimos que para todo $0 \leq k \leq n-1$, se tiene que $\delta_n(s,v) \leq \delta_k(s,v)$. Sigue que

$$\max_{0\leq k\leq n-1} \frac{\delta_n(s,v)-\delta_k(s,v)}{n-k} \quad \leq \quad 0\,.$$

La otra desigualdad se tiene de (ii).

(v).- Si incrementamos en t el peso de cada arco en G, entonces

(1) el peso promedio de cada ciclo aumenta en t, luego μ^* aumenta en t,

(2) el valor de $\delta_k(s,v)$ se incrementa en kt cualquiera sea $k \in \mathbb{N}$ y $v \in V$. Luego, el valor de $(\delta_n(s,v)-\delta_k(s,v))/(n-k)$ se incrementa en t.

Denotemos por $\widetilde{\mu}^*$ (respectivamente $\widetilde{\delta}_k(u,v)$) el peso de un ciclo de peso promedio mínimo (respectivamente el peso de un camino de largo mínimo de u a v que ocupa exactamente k arcos) en el grafo que se obtiene a partir de G incrementando el peso de cada arco en $-\mu^*$. Sigue que $\widetilde{\mu}^* = 0$. Luego, por (ii) y (iv),

$$0 \;=\; \min_{v\in V}\max_{0\leq k\leq n-1} \frac{\widetilde{\delta}_n(s,v)-\widetilde{\delta}_k(s,v)}{n-k} \;=\; \min_{v\in V}\max_{0\leq k\leq n-1} \frac{\delta_n(s,v)-\delta_k(s,v)}{n-k} - \mu^*\,.$$

Despejando se obtiene la conclusión deseada.

(vi).- La parte anterior sugiere utilizar el algoritmo de la Fig. 2 para calcular μ^*.

ENTRADA: $G=(V,E)$, $s \in V$, y $w{:}E \to \mathbb{R}$.
CÓDIGO:
for all $k=0$ **to** $|V|$ **do**
  for all $v \in V$ **do** $d_k(v) \leftarrow \infty$,
$d_0(s) \leftarrow 0$
for $k=1$ **to** $|V|$ **do**
  for all $(u,v) \in E$ **do**
    if $d_k(v) > d_{k-1}(u)+w(u,v)$
      then $d_k(v) \leftarrow d_{k-1}(u)+w(u,v)$
$min \leftarrow +\infty$
for all $v \in V$ **do**
  $max \leftarrow -\infty$
  for all $k=0$ **to** $|V|$ **do**
    if $max < (\delta_n(s,v)-\delta_k(s,v))/(n-k)$
      then $max \leftarrow (\delta_n(s,v)-\delta_k(s,v))/(n-k)$
  if $min > max$ **then** $min \leftarrow max$
return min

Figure 2: Algoritmo para calcular μ^*.