MA4701-1 - Optimización Combinatorial
Profesor: José Soto
Auxiliares: Nicolás Sanhueza - Christian von Borries

Solución a Control N°1

14 de septiembre de 2013

P1. (1 punto) Sean $G = (V, E)$ grafo conexo, $w : E \to \mathbb{R}^+$ estrictamente positiva y $\mathcal{X}$ el conjunto de los árboles generadores de G.

(a) $(0,5$ puntos) Demuestre que $T \in \mathcal{X}$ minimiza $w(E(T)) = \sum_{e \in E(T)} w(e)$ si y solo si minimiza $\prod_{e \in E(T)} w(e)$.

sol En primer lugar, notemos que

$$\prod_{e \in E(T)} w(e) = \exp\left(\ln\left(\prod_{e \in E(T)} w(e)\right)\right) = \exp\left(\sum_{e \in E(T)} \ln(w(e))\right)$$

y como la función $\exp : \mathbb{R} \to \mathbb{R}$ es estrictamente creciente, basta demostrar que un árbol generador es de peso mínimo para w si y sólo si lo es para la función de pesos $\ln \circ w$, la composición del logaritmo y w.

Supongamos que $E(G) = \{e_1, e_2, \ldots, e_m\}$ y que, dada la función de pesos w, se ordenan en forma creciente

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$$

Como la función $\ln : \mathbb{R}^+ \to \mathbb{R}$ es estrictamente creciente, se tiene por lo tanto que

$$\ln(w(e_1)) \leq \ln(w(e_2)) \leq \cdots \leq \ln(w(e_m))$$

Es decir, las aristas tienen exactamente el mismo orden bajo las funciones de peso w y $\ln \circ w$. Como el algoritmo de Kruskal ordena las aristas de acuerdo a su peso y las procesa en ese orden, escogerá las mismas aristas en ambos casos. Las aristas escogidas por el algoritmo formarán un árbol generador óptimo (en ambos casos) por la correctitud del algoritmo; así que el mismo árbol minimiza ambas funciones de pesos, que es lo queríamos demostrar. ■

(b) $(0,5$ puntos) Demuestre o dé un contraejemplo: $\mathcal{X}$ tiene un único mínimo si para cada ciclo $C \subseteq E$, todos los pesos de C son distintos.

sol La proposición es cierta. Para demostrarlo, supongamos que no lo es. Sea G un grafo y $w : E(G) \to \mathbb{R}$ una función de pesos en las aristas de G tal que en todo ciclo de G, todo par de aristas tiene pesos distintos. Si la proposición no es cierta, existen T_1, T_2 árboles generadores de peso mínimo distintos.

Como T_1 y T_2 son distintos, $E(T_1)\Delta E(T_2)$ no es vacío. Sea $e \in E(T_1)\Delta E(T_2)$ tal que $w(e)$ es mínimo. Sin pérdida de generalidad, $e \in E(T_1)$.

Como $e \notin E(T_2)$, $T_2 + e$ tiene un ciclo que llamamos C. Notemos que para toda $f \in C$, $w(f) \leq w(e)$ (si no, $T_2 - f + e$ sería un árbol de peso estrictamente inferior al mínimo). Como en todo ciclo los pesos son distintos, se tiene que para toda $f \in C - e$ la desigualdad es estricta, $w(f) < w(e)$.

Notemos finalmente que $C - e \nsubseteq E(T_1)$, pues si no, $C - e + e = C \subseteq T_1$ y un árbol no puede tener ciclos. Entonces existe una arista e' en $C - e$ que está en $E(T_2)$ pero no en $E(T_1)$, luego, está en $E(T_1) \Delta E(T_2)$. Por lo observado anteriormente, $w(e') < w(e)$, lo que contradice la minimalidad de $w(e)$. $\blacksquare$

P2. (1 punto + $0,5$ extra). Dado un grafo $G = (V, E)$ conexo y $w : E \to \mathbb{R}$ una función de peso, sea T_1 un árbol generador de peso mínimo de G. Un árbol generador T_2 se llama *segundo mínimo* de G si $w(E(T_1)) < w(E(T_2))$ y no existe árbol generador T' con $w(E(T_1)) < w(E(T')) < w(E(T_2))$.

Diseñe un algoritmo que encuentre un segundo mínimo de G (si no existe, el algoritmo debe reportarlo). Analice la correctitud y complejidad de su algoritmo.

Indicación: Pueden haber varias aristas con el mismo peso. Le puede servir probar que si existe segundo mínimo, entonces existe un par (T_1, T_2) con T_1 mínimo y T_2 segundo mínimo con muchas aristas en común.

Nota: El puntaje del Problema 2 dependerá de la complejidad de su algoritmo (los puntajes no se acumulan).

- Si prueba que es correcto y polinomial, obtiene: $0,5$ puntos.
- Si prueba que es correcto y corre en $O(|E||V|)$, obtiene puntaje completo: 1,0 puntos.
- Si prueba que es correcto y corre en $O(|V|^2)$, obtiene puntaje extra: 1,5 puntos.

<u>**sol**</u> Probaremos primero la indicación. Sea T_1 árbol generador de peso mínimo y supongamos que existe un árbol generador que es segundo mínimo. Afirmamos que existe un árbol generador T_2 que es segundo mínimo tal que $|T_1 \triangle T_2| = 2$ (o sea, tienen muchos arcos en común). En efecto, sea T_2 árbol generador que es segundo mínimo tal que $|T_1 \triangle T_2|$ es mínimo. Tomemos $e = uv \in T_2 \backslash T_1$. Como T_1 es árbol generador de peso mínimo, necesariamente $w(f) \leq w(e)$ para cualquier arco f en el (u, v)-camino de T_1. Sea f un arco del (u, v)-camino de T_1 tal que $f \notin T_2$ (existe, ya que si no T_2 tendría un ciclo). Tenemos dos casos:

- Caso 1: $w(f) = w(e)$. Entonces si tomamos $T_2' = T_2 + e - f$, T_2' es un nuevo árbol generador y $w(T_2') = w(T_2)$, o sea es segundo mínimo. Además $|T_1 \triangle T_2'| = |T_1 \triangle T_2| - 2$, contradiciendo que T_2 era un árbol generador que era segundo mínimo que minimiza $|T_1 \triangle T_2|$.
- Caso 2: $w(f) < w(e)$. Sea $T_2' = T_2 + f - e$. Entonces $w(T_2') < w(T_2)$, luego es árbol generador de peso mínimo, o sea $w(T_1) = w(T_2')$. Sea ahora $T_1' = T_1 + e - f$. Entonces T_1' es árbol generador que es segundo mínimo, pues $w(T_1') = w(T_1) + w(e) - w(f) = w(T_2)$. Además $|T_1 \triangle T_1'| = 2$. Como T_2 minimiza $|T_1 \triangle T_2|$, esto último implica que $|T_1 \triangle T_2| = 2$.

Esto último nos dice que T_2 es del tipo $T_1 + e - f$, con $e \notin T_1$ y $f \in T_1$. Nuestro algoritmo probará entonces agregando todos los arcos $e = uv \notin T_1$ posibles. Sin embargo, en vez de probar quitando cualquier arco de T_1 (esto sería polinomial pero no muy eficiente), basta probar quitando un arco f del (u, v)-camino ya que de otra forma $T_1 + e - f$ tendría un ciclo. Además, como estamos buscando un segundo mínimo, basta probar quitando el arco f que minimice $\{w(e) - w(f) : w(e) - w(f) > 0\}$.

Entonces, el algoritmo que usaremos es el siguiente:

Algoritmo 1 Encontrar árbol generador segundo mínimo

Encontrar T_1 árbol generador de peso mínimo.
for $e = uv \notin T_1$ **do**
  Encontrar f en el (u,v)-camino de T_1 que minimice $w(e) - w(f)$ con $w(e) \neq w(f)$.
  Ir guardando el par (e', f') que minimice $w(e') - w(f')$.
end for
if Nunca se guardó un par (e', f') (o sea siempre se cumplía que $w(e) = w(f)$) **then**
  return No existe árbol generador que es segundo mínimo.
else
  return $T_1 + e' - f'$
end if

Si existe un segundo mínimo, el razonamiento anterior nos dice que existe un segundo mínimo de la forma $T_1 + e - f$. Como elegimos el par (e, f) tal que $T_1 + e - f$ es árbol generador y minimiza $w(e) - w(f)$, $T_1 + e - f$ necesariamente es segundo mínimo.

Por otro lado, si no existe un árbol generador que sea segundo mínimo, todos los árboles generadores son mínimos. Entonces, para cualquier par (e, f) tal que $T_1 + e - f$ es árbol generador, $w(T_1) = w(T_1 + e - f)$. Luego $w(e) = w(f)$ para cualquier par (e, f) que considere el algoritmo y por lo tanto el algoritmo retorna correctamente que no existe árbol generador que es segundo mínimo.

Analicemos ahora la complejidad del algoritmo. Encontrar T_1 árbol generador de peso mínimo lo podemos hacer en $O(|E| + |V| \log |V|)$.

Al ciclo for entraremos $O(|E|)$ veces. Dado $e = uv$ encontrar el arco más pesado del (u, v)-camino en T_1 se puede hacer con un DFS en T_1 partiendo de u y guardando el arco de mayor peso del camino que está recorriendo DFS. Cuando lleguemos a v, habremos encontrado f en el (u, v) camino de T_1 tal que $w(e) - w(f)$ es mínimo. Este DFS demora $O(|V| + |E(T_1)|) = O(|V|)$. Luego, en total el algoritmo es $O(|V||E|)$.

Para hacerlo en $O(|V|^2)$, primero lo haremos suponiendo que los pesos de todos los arcos son distintos (cuando hay arcos del mismo peso es parecido pero con pequeñas modificaciones que diremos después). Podemos definir $A[u, v]$ como el arco de peso máximo del (u, v)-camino de T_1. Suponiendo que tenemos esto calculado, los pasos dentro del ciclo for se pueden hacer en $O(1)$: Como los pesos son todos distintos, no hay que preocuparse del caso en el que $w(e) = w(f)$. Luego, dado $e = uv$, basta tomar $f = A[u, v]$. Calcular $A[u, v]$ para todo u y para todo v se puede hacer en $O(|V|^2)$. De hecho, para u fijo $A[u, v]$ se puede puede calcular en $O(|V|)$ para todo v, haciendo un DFS en T_1 partiendo de u y guardando el arco más pesado en el camino que lleva el DFS. Como es un DFS en un árbol, éste demora $O(|V|)$. Repitiendo esto para todos los vértices, calculamos $A[u, v]$ en $O(|V|^2)$. Pondremos un pseudocódigo por claridad:

Antes de iniciar el algoritmo:
$V(T) \leftarrow \emptyset$
$E(T) \leftarrow \emptyset$
DFS-GuardaArcoMáximo(u, T_1, u)

Algoritmo 2 DFS-GuardaArcoMáximo(u, T_1, v) (Vértice de inicio, árbol en el que se está ejecutando el DFS, vértice en el que va el DFS)

$V(T) \leftarrow V(T) \cup v$
for $x \in N(v)$ **do**
  if $x \notin V(T)$ **then**
    $E(T) \leftarrow E(T) \cup vx$
    if $w(A[u, v] > w(vx]))$ **then**
      $A[u, x] = vx$
    else
      $A[u, x] = A[v, x]$
    end if
    DFS-GuardaArcoMáximo (u, T_1, v)
  end if
end for

Si hay arcos que pueden ser iguales, podemos adaptar la idea anterior. En vez de guardar el arco más pesado del (u, v)-camino de T_1 podemos guardar el arco más pesado y el segundo más pesado. Al agregar $e = uv$ hay que comparar: Si el arco más pesado del (u, v)-camino tiene menor peso que e, tomar f como el arco más pesado, si no tomar f como el segundo arco más pesado.

En total tenemos entonces:

- Encontrar un árbol generador de peso mínimo T_1. ($O(|E|+|V| \log |V|)) = O(|V|^2)$).
- Precalcular, para cada par (u, v), el arco más pesado y el segundo más pesado del (u, v)-camino en T_1. ($O(|V|^2)$)
- Entrar al for $O(|E|) = O(|V^2|)$ veces. Cada paso dentro del ciclo se ejecuta en $O(1)$.

Sumando, el orden del algoritmo queda $O(|V|^2)$. ■

P3. (2 puntos)

(a) (0,6 puntos) [Imagen de una matroide es matroide]

Sea $M' = (E', \mathcal{I}')$ una matroide, E un conjunto finito tal que $E \cap E' = \emptyset$ y $\phi : E' \to E$ una función cualquiera. Definimos $M := \phi(M) = (E, \mathcal{I})$, con $\mathcal{I} = \{\phi(X) : X \in \mathcal{I}'\}$. Demuestre que M es una matroide.

Indicación: Para probar que M satisface el axioma de aumento para $X, Y \in \mathcal{I}$, elija conjuntos X' e Y' en $\mathcal{I}'$ con $\phi(X') = X$, $\phi(Y') = Y$, $|X'| = |X|$, $|Y'| = |Y|$ y de modo que $|X' \cap Y'|$ sea lo más grande posible.

sol Notemos en primer lugar que $\phi(\emptyset) = \emptyset$, así que $\emptyset \in \mathcal{I}$. Además, si $Y \in \mathcal{I}$ y $X \subseteq Y$, sea $Y' \in \mathcal{I}$ tal que $\phi(Y') = Y$. Notemos que $\phi^{-1}(X) \cap Y'$ tiene por imagen a X. Además, $\phi^{-1}(X) \cap Y' \subseteq Y' \in \mathcal{I}'$, así que $\phi^{-1}(X) \cap Y' \in \mathcal{I}'$, por lo tanto $X \in \mathcal{I}$. Esto demuestra que M es par hereditario, basta ver que cumple algún axioma de aumento para concluir que es matroide. Demostraremos aumento fuerte.

Notemos que siempre podemos escoger preimágenes del mismo tamaño que la imagen, pues dada cualquier función entre conjuntos finitos tal que $\phi(A') = A$, para cada elemento $a \in A$ podemos escoger una preimagen $a' \in A'$. El conjunto de todas estas preimágenes escogidas entrega un conjunto A'' tal que $\phi(A'') = A$ y $|A''| = |A|$. Notemos además que, por cardinalidad, la función ϕ restringida a A'' es biyección.

Seguimos, entonces, la indicación. Sean $X, Y \in \mathcal{I}$, y conjuntos X' e Y' en $\mathcal{I}'$ con $\phi(X') = X$, $\phi(Y') = Y$, $|X'| = |X|$, $|Y'| = |Y|$ y de modo que $|X' \cap Y'|$ sea lo más grande posible.

Por un lado, $M' = (E', \mathcal{I}')$ es matroide, así que como $|X'| = |X| < |Y| = |Y'|$, se tiene que existe un $e' \in Y' \setminus X'$ tal que $X' + e' \in \mathcal{I}'$. Si la imagen de e' bajo ϕ cae fuera de X, entonces $\phi(X' + e') = \phi(X') + \phi(e') = X + e$ para algún $e \in Y \setminus X$ y por lo tanto, $X + e \in \mathcal{I}$; así que en este caso se concluye. Supongamos, entonces, que $\phi(e) \in X$.

Como la restricción a X' dada por $\phi : X' \to X$ es biyección, existe un x' tal que $\phi(x') = \phi(e')$. Además este x' no pertenece a Y', pues la restricción de ϕ a Y' dada por $\phi : Y' \to Y$ es biyección.

Consideremos entonces el conjunto $X'' := X' + e' - x'$. Como $X'' \subseteq X' + e' \in \mathcal{I}'$, se tiene que $X'' \in \mathcal{I}'$. Además, se tiene que $\phi(X'') = \phi(X' + e' - x') = \phi(X') + \phi(e') - \phi(x') = X$; y además $|X''| = |X'| + |e'| - |x'| = |X'| = |X|$.

Por último, notemos que $|X'' \cap Y'| = |(X' + e' - x') \cap Y'| = |(X' + e') \cap Y'| = |X' \cap Y'| + 1 > |X' \cap Y'|$, lo que es una contradicción pues X' e Y' fueron escogidos para que esta intersección fuera lo más grande posible. Por lo tanto este caso no puede pasar, y se concluye. ∎

(b) $(0,4$ puntos) [Unión de matroides es matroide]

Sean $M_1 = (E_1, \mathcal{I}_1)$ y $M_2 = (E_2, \mathcal{I}_2)$ matroides donde E_1 y E_2 pueden intersectarse. Usando la parte anterior demuestre que

$$M_1 \vee M_2 := (E_1 \cup E_2, \{F_1 \cup F_2 : F_1 \in \mathcal{I}_1, F_2 \in \mathcal{I}_2\})$$

es una matroide.

sol Supongamos que $M_1 = (E_1, \mathcal{I}_1)$ y $M_2 = (E_2, \mathcal{I}_2)$, donde E_1 y E_2 no son necesariamente disjuntos. Definamos $E := E_1 \cup E_2$ y consideremos para cada M_1 y M_2 las funciones «etiquetadoras» definidas por

$$\begin{array}{rccc} \ell_1 : & E_1 & \to & E \times \{1\} \\ & x & \mapsto & (x,1) \end{array} \qquad \begin{array}{rccc} \ell_2 : & E_2 & \to & E \times \{2\} \\ & x & \mapsto & (x,2) \end{array}$$

Estas funciones son biyecciones entre sus recorridos y sus imágenes, y definen de manera natural matroides en $E \times \{1\}$ y $E \times \{2\}$, dadas por las imágenes de los conjuntos independientes en M_1 y M_2. Llamemos a estas matroides M_1' y M_2'.

Los conjuntos $E \times \{1\}$ y $E \times \{2\}$ son disjuntos; pues el primero contiene sólo tuplas cuya segunda coordenada es 1, y el segundo sólo tuplas cuya segunda coordenada es 2. Esto permite definir la suma directa de las matroides M_1' y M_2', que denotamos $M_1' \oplus M_2'$ y tiene por conjunto base a $E \times \{1, 2\}$.

Finalmente, consideremos la función «desetiquetadora» ϕ, que definimos como

$$\begin{array}{rccc} \phi : & E \times \{1,2\} & \to & E_1 \cup E_2 \\ & (x,i) & \mapsto & x \end{array}$$

Por la parte anterior, las imágenes de los conjuntos independientes en $M_1' \oplus M_2'$ formarán una matroide en $E_1 \cup E_2$, llamémosla $\overline{M}$. Basta probar que esta matroide es igual a $M_1 \vee M_2$.

Efectivamente es así. Sea X un independiente en $\overline{M}$. Por definición, X es imagen bajo ϕ de un independiente en $M_1' \oplus M_2'$, pero en la suma directa los independientes son uniones disjuntas de independientes en cada uno de sus componentes, así que $X = \phi(X_1' \cup X_2')$ para X_1' independiente en M_1' y X_2' independiente en M_2'. Por lo anteriormente observado, un independiente en M_1' es solamente una versión «etiquetada» de un independiente en M_1, lo mismo con M_2'; y como la función ϕ elimina la segunda coordenada de las tuplas que se definieron, tenemos que $X = X_1 \cup X_2$ para X_1 independiente en M_1 y X_2 independiente en M_2. Análogamente, si X es un conjunto de la familia definida en $M_1 \vee M_2$, es unión de independientes X_1 y X_2 en M_1 y M_2, respectivamente; y en este caso basta considerar $\phi(\ell_1(X_1) \cup \ell_2(X_2))$ para ver que está incluído en la matroide $\overline{M}$. Entonces $\overline{M} = M_1 \vee M_2$ y se concluye. ■

(c) (0,4 puntos) Sea $M = (E, \mathcal{I})$ una matroide y e, f elementos de E. Decimos que e y f en E son *paralelos* si $\{e, f\}$ es un circuito de M ó $e = f$. Decimos que e y f en E son *seriales* si $\{e, f\}$ es un cocircuito de M (i.e. circuito de M^*), ó $e = f$. Demuestre que «ser paralelos» y «ser seriales» son relaciones de equivalencia en E.

sol Demostraremos que «ser paralelos» es relación de equivalencia.

Que «ser paralelo» es reflejo es fácil, pues dado cualquier $e \in E$, se tiene que $e = e$. Ser simétrico también es fácil, pues si e y f son paralelos, o bien $e = f$, o bien $\{e, f\}$ es circuito de M. En ambos casos es evidente que el orden en que consideremos e y f da lo mismo. Por lo tanto, sólo queda demostrar transitividad.

Sean (e, f) y (f, g) pares de elementos paralelos. Si $e = f$ ó $f = g$ ó $e = g$ entonces verificar la paralelidad de e y g es fácil. Entonces basta verificar el caso en que $\{e, f\}$ y $\{f, g\}$ son circuitos. Como los circuitos son conjuntos dependientes minimales, se infiere que $\{e\}$, $\{f\}$, $\{g\} \in \mathcal{I}$.

Supongamos que (e, g) no es un par paralelo. Como no son iguales, entonces debe suceder que $\{e, g\}$ no es circuito. Por lo dicho anteriormente, todo subconjunto propio de $\{e, g\}$ es independiente; así que la única forma en que $\{e, g\}$ puede fallar de ser un circuito es que $\{e, g\} \in \mathcal{I}$. Notemos que $\{f\}, \{e, g\} \in \mathcal{I}$ y que $|\{f\}| < |\{e, g\}|$, así que por axioma de aumento, debe existir un $x \in \{e, g\}$ tal que $\{f, x\} \in \mathcal{I}$. Pero cualquiera de las posibles elecciones de x, junto a f, forma un conjunto que sabemos que es circuito; lo que es una contradicción. Por lo tanto, (e, g) son paralelos. Luego, «ser paralelos» es relación de equivalencia.

Finalmente, dada una matroide M, «ser paralelos» es una relación de equivalencia en las aristas de la matroide dual, y los circuitos de M^* se corresponden con los cocircuitos de M; así que se deduce que «ser seriales» es relación de equivalencia en E. ∎

(d) (0,6 puntos) Sea $M = (E, \mathcal{I})$ una matroide que no posee cocircuitos de cardinalidad 1 (coloops). Dos elementos e y f en E se dicen *similares* si la función $\phi : E \to E$ que intercambia e con f; es decir $\phi(e) = f$, $\phi(f) = e$ y $\phi(x) = x$, para todo $x \notin \{e, f\}$; es un automorfismo de M. En otras palabras, e y f son similares si se cumple que

$$X \in \mathcal{I} \iff \phi(X) \in \mathcal{I}$$

Demuestre que e y f son similares si y solo si $(M/e) \setminus f = (M/f) \setminus e$.

<u>**sol**</u> Antes que nada, notemos que siempre podemos considerar e y f distintos, pues si son iguales es todo trivial. Luego notemos que, por caracterizaciones conocidas, dada una matroide $M = (E, \mathcal{I})$ y $e, f \in E$ distintos, tenemos que

$$\mathcal{I}_e := \mathcal{I}((M/e) \setminus f) = \{X \in \mathcal{I} : X \subseteq E \setminus \{e, f\},\ X + B_e \in \mathcal{I}\}$$

donde B_e es una base de $\{e\}$. Como $\{e\}$ es síngleton, sólo hay dos posibilidades para una base: $\{e\}$ mismo o el vacío, que se tendran en los casos $\{e\} \in \mathcal{I}$ y $\{e\} \notin \mathcal{I}$ respectivamente.

$\Rightarrow$) Si e y f son similares, tenemos que $\{e\} \in \mathcal{I}$ si y sólo si $\{f\} \in \mathcal{I}$. Por la caracterización de la base de un síngleton notada antes, dado cualquier $X \subseteq E \setminus \{e, f\}$ se tiene que $X + B_e \in \mathcal{I}$ si y sólo si $X + B_f \in \mathcal{I}$. Dada la caracterización de los independientes de $(M/e) \setminus f$ mencionada anteriormente, esto basta para decir que $(M/e) \setminus f = (M/f) \setminus e$.

$\Leftarrow$) Basta probar que para todo $X \in \mathcal{I}$, $\phi(X) \in \mathcal{I}$, pues para la otra implicancia basta aplicar ϕ de nuevo, que compuesta dos veces consigo misma es la identidad. Si $\{e, f\} \cap X = \emptyset$ o $\{e, f\} \subseteq X$, se tiene que $\phi(X) = X$ y no hay nada que probar, así que los únicos casos interesantes son cuando un elemento de $\{e, f\}$ está en X y otro no.

Por otro lado, notemos que si $\{e\} \notin \mathcal{I}$ y $\{f\} \notin \mathcal{I}$, ningún $X \in \mathcal{I}$ puede contener a ninguno de los dos (por propiedad hereditaria) y por lo tanto este caso no tiene nada de interesante.
Si es que $\{e\} \notin \mathcal{I}$ y $\{f\} \in \mathcal{I}$, se tiene que $B_e = \emptyset$ y $B_f = \{f\}$. Esto implica que

$$\mathcal{I}_e = \{X \in \mathcal{I} : X \subseteq E \setminus \{e, f\},\ X \in \mathcal{I}\}$$
$$\mathcal{I}_f = \{X \in \mathcal{I} : X \subseteq E \setminus \{e, f\},\ X + f \in \mathcal{I}\}$$

Por hipótesis, $\mathcal{I}_e = \mathcal{I}_f$. Esto implica que si $X \in \mathcal{I}$ no contiene a f, $X + f \in \mathcal{I}$. Luego, todo independiente maximal contiene a f, así que f está contenido en toda base de M. Luego $\{f\}$ es cocircuito de cardinalidad uno, lo que es imposible por hipótesis. Análogamente, este razonamiento descarta el caso en que $\{e\} \in \mathcal{I}$ y $\{f\} \notin \mathcal{I}$.
Sólo queda el caso en que $\{e\}, \{f\} \in \mathcal{I}$ y X es tal que contiene a sólo uno de los dos. Supongamos sin pérdida de generalidad que $e \in X$ y $f \notin X$. Entonces $X - e \in \mathcal{I}_e = \mathcal{I}_f$, así que $X - e + f \in \mathcal{I}$. Pero justamente $\phi(X) = X - e + f$, así que se concluye. ■

P4. (2 puntos+0,5 extra) Sea $G = (V, E)$ un grafo dirigido simple. Es decir, V es un conjunto finito de nodos y $E \subseteq V \times V$ es un conjunto de arcos. Si $e = (v, w)$ decimos que e está dirigido hacia w. Sea $c : V \to \mathbb{N}$ una función de capacidad en los nodos y $w : E \to \mathbb{R}$ una función de peso en los arcos. Decimos que un conjunto de arcos $F \subseteq E$ es c-convergente si para todo vértice v hay a lo más $c(v)$ arcos en F dirigidos hacia v.

(a) (0.5 puntos). Definimos $Z(G, c)$ como la familia de conjuntos c-convergentes de G. Pruebe que $(E, Z(G, c))$ es una matroide.

<u>sol</u> Sea $I_v = \{X \subseteq V : \forall e \in X \; e$ está dirigido hacia v y $|X| \leq c(v)\}$. Entonces definiendo $M_v = (E, I_v)$ es una matroide uniforme y $(E, Z(G, c)) = \bigoplus_{v \in V} M_v$, luego $(E, Z(G, c))$ es matroide. ■

(b) (1 punto+0.5 extra). Para $t \in \mathbb{N}$ sea $Z_t(G, c)$ la familia de conjuntos c-convergentes de G con exactamente t arcos. Diseñe un algoritmo que encuentre $X \in Z_t(G, c)$ con $w(X)$ mínimo (si tal X no existe su algoritmo debe reportarlo). Analice la correctitud y complejidad de su algoritmo.

Nota: El puntaje de esta parte dependerá de la complejidad de su algoritmo (Los puntajes no se acumulan)

- Si prueba que es correcto y polinomial, obtiene 0,5 puntos.
- Si prueba que es correcto y es $O(|E| \log |E| + |V|)$, obtiene puntaje completo: 1,0 puntos.
- Si prueba que es correcto y es $O(|E| + |V| \log |V|)$, obtiene puntaje extra: 1,3 puntos.
- Si prueba que es correcto y es $O(|E| + |V|)$, obtiene puntaje extra: 1,5 puntos. Esto puede requerir conocimiento de algoritmos externos al curso.

<u>sol</u> Basta encontrar una base de peso mínimo de la matroide truncada $(E, Z(G, c))^t$ y revisar si tiene tamaño t. Usaremos el algoritmo GLOTÓN-BASE visto en clases. Ordenar los arcos toma $|E| \log |E|$. Llamemos S al conjunto que es construido por GLOTÓN-BASE. Dado e, revisar si $S + e$ es independiente lo podemos hacer en $O(1)$ si guardamos un contador de arcos entrantes de S para cada vértice. Luego podemos implementar GLOTÓN-BASE en $O(|E| \log |E| + |V|)$.

Para hacerlo más rápido, podemos encontrar primero para cada v una base de peso mínimo de M_v (definido en la parte a) que llamaremos B_v. Para ello usaremos el algoritmo QUICKSELECT (QUICKSELECT encuentra en un arreglo de tamaño n el k-ésimo menor elemento en tiempo $O(n)$). En pseudocódigo, el algoritmo es el siguiente:

Algoritmo 3

for $v \in V$ **do**
  $q(v) \leftarrow$el $c(v)$-ésimo menor elemento de los pesos de $\delta^-(v)$ (Encontrado con QuickSelect)
  $B_v \leftarrow \emptyset$
  for $e \in \delta^-(v)$ **do**
    if $w(e) \leq q(v)$ **then**
      $B_v \leftarrow B_v + e$
    end if
  end for
end for
$B := \bigcup_{v \in V} B_v$
if $|B| < t$ **then**
  return $Z_t(G, c) = \emptyset$
end if
$q \leftarrow$el t-ésimo menor elemento de los pesos de E (Encontrado nuevamente con QuickSelect)
$S \leftarrow \emptyset$
for $e \in E$ **do**
  if $w(e) \leq q$ **and** $|S| < t$ **then**
    $S \leftarrow S + e$
  end if
end for

Como $(E, Z(G, c))$ es una matroide partición, el conjunto B definido antes es una base de $(E, Z(G, c))$ de peso mínimo. Después el algoritmo toma los t menores elementos de B: Esto necesariamente es un conjunto independiente de peso mínimo de tamaño t, pues no es más que usar Glotón-Base en la matroide truncada.

Calculemos el orden del algoritmo. Dado v, con el método anterior demoramos $O(1 + |\delta^-(v)|)$ en encontrar B_v. En total nos demoramos entonces $\sum_{v \in V} O(1 + |\delta^- v|)$, que es $O(|V| + |E|)$ por el "Handshaking Lemma". El resto del algoritmo demora $O(|E|)$, dando en total $O(|V| + |E|)$.

■

(c) (0,5 puntos). Sea $G = (V, E)$ un grafo simple no dirigido ($E \subseteq \binom{E}{V}$). Definimos el par

$$Q = (E, \{F \subseteq E : \text{ cada componente conexa de } (V, F) \text{ contiene a lo más un ciclo}\})$$

Demuestre que Q es una matroide.

Indicación: Pruebe que Q es imagen (ver Problema 3) de una matroide como las definidas en la parte **4(a)**.

sol Sea $G' = (V, E')$ el grafo dirigido definido por $E' = \{(u, v) : uv \in E\}$. Además tomamos $c(v) = 1$ y definimos

$$\begin{aligned} f : E' &\to E \\ (u, v) &\mapsto f(u, v) = uv \end{aligned}$$

Entonces Q es imagen vía f de la matroide $Z(G', c)$. O sea F es independiente en $Q \iff$ existe $X \in Z(G', c)$ tal que $F = f(X)$.

En efecto, sea X un conjunto c-convergente de G'. Sin pérdida de generalidad, $f(X)$ es conexo, si no basta descomponer en componentes conexas. Sea V' el conjunto de nodos adyacentes a X. Primero que nada, notar que como en cada nodo entra a lo más un arco, entonces hay a lo más $|V'|$ arcos en X. Entonces $|f(X)| \leq |X| \leq |V'|$. Por otro lado, por conexidad tenemos que $|V'| - 1 \leq |f(X)|$. Si $|f(X)| = |V'| - 1$, entonces $(V', f(X))$ es un árbol. Si no, $|f(X)| = |V'|$ y Como $f(X)$ es conexo contiene un árbol de $|V'| - 1$ arcos y agregando el arco que falta se forma un único ciclo.

Por otro lado, si F es independiente de Q, podemos orientar los arcos de cada componente conexa de tal forma que cada nodo tenga a lo más un arco dirigida hacia él. Si la componente conexa tiene un ciclo, podemos dirigir el ciclo completo en una dirección (hay dos direcciones posibles, da lo mismo cuál tomar). Si quitamos los arcos del ciclo, entonces queda un bosque (pues eliminamos el ciclo). Sea T un árbol de este bosque y sea v el vértice de este ciclo. Recorriendo los nodos de T en orden de un DFS desde v y orientando los arcos en ese orden, obtenemos lo pedido. En efecto, como estamos haciendo DFS, cada nodo es recorrido exactamente una vez, luego entra a lo más un arco a cada nodo. (Nota: Es importante comenzar desde v, ya que ya tiene un arco entrante por el ciclo.) Además, como T es árbol el DFS orientará todos los arcos. Si la componente conexa no tiene un ciclo, entonces es un árbol y podemos orientar los arcos del árbol de la misma forma. ■