MA3705. Algoritmos Combinatoriales 2020.
Profesor: José Soto
Profesores Auxiliares: Antonia Labarca, Víctor Saez

Solución tarea 1.

Problema 1. Colaboración autorizada (8 puntos)

Sea $(S, \mathcal{B})$ un clutter no vacío que satisface el axioma de intercambio (es decir, satisface las propiedades de la sección 3.1 de [1]). Defina

$$\mathcal{I} = \{X \subseteq S\colon\ \exists B \in \mathcal{B}, X \subseteq B\}$$
$$\mathcal{B}' = \{X \in \mathcal{I}\colon\ \forall e \in S \setminus X, X + e \notin \mathcal{I}\}$$

Demuestre que $(S, \mathcal{I})$ es el sistema de independencia de una matroide (es decir, satisface cualquiera de las definiciones 2.1; 2.2 o 2.3 de [1]) y que su conjunto de bases $\mathcal{B}'$ satisface $\mathcal{B}' = \mathcal{B}$.

Solución:

Como $\mathcal{B}$ es no vacío, existe $B \in \mathcal{B}$ y como $\emptyset \subseteq B$, $\emptyset \in \mathcal{I}$. Por otro lado si $X \subseteq Y \in \mathcal{I}$, existe $B \in \mathcal{B}$ tal que $Y \subseteq B$ y luego $X \subseteq B$, por lo que $X \in \mathcal{I}$. Esto prueba que $(S, \mathcal{I})$ es sistema de independencia.

Probemos por inducción en $k = |B \setminus B'|$ que si B y B' están en $\mathcal{B}$ entonces $|B| = |B'|$. Si $k = 0$, entonces $B \subseteq B'$. Como $\mathcal{B}$ es clutter, $B = B'$. Si $k \geq 1$ entonces podemos tomar $e \in B \setminus B'$. Por intercambio existe $f \in B' \setminus B$ tal que $B + e - f \in \mathcal{B}$, pero $|B + e - f \setminus B'| = k - 1$ de lo que se concluye por hipótesis de inducción que $|B + e - f| = |B'|$. Pero entonces $|B| = |B + e - f| = |B'|$, terminando la demostración de este resultado intermedio.

Probemos que $\mathcal{I}$ satisface aumento débil. Sean X, $Y \in \mathcal{I}$ con $Y \setminus X = \{a, b\}$, $X \setminus Y = \{c\}$. Elijamos $B_X, B_Y \in \mathcal{B}$ tal que $X \subseteq B_X, Y \subseteq B_Y$, con $|B_X \cap B_Y|$ máximo. Observemos que $a, b \in B_Y$. Supongamos por contradicción que ambos elementos no están en B_X. Es decir, $\{a, b\} \subseteq B_Y \setminus B_X$. Como $|B_X| = |B_Y|$ deben existir al menos dos elementos en $B_X \setminus B_Y$, y como no pueden ser ambos iguales a c, existe $d \in B_X \setminus B_Y$ con $d \notin X \cup Y$. Por intercambio, existe $e \in B_Y \setminus B_X$ tal que $B_X - d + e \in \mathcal{B}$. Pero $B_X - d + e$ contiene a X y $|(B_X - d + e) \cap B_Y| = |B_X \cap B_Y| + 1$ contradiciendo la elección de B_X y B_Y. Luego, uno de a o b están en B_X y entonces $X + a$ o $X + b$ es independiente.

Para finalizar, sea $\mathcal{B}'$ el conjunto de bases de $(S, \mathcal{I})$ y sea k el cardinal común de los elementos del clutter $\mathcal{B}$. Sea $B \in \mathcal{B}$. Notemos que $B \in \mathcal{I}$ pues $B \subseteq B \in \mathcal{B}$. Por otro lado si $e \in S \setminus B$, $|B + e| = k + 1$ por lo que $B + e$ no puede ser subconjunto de ningún elemento de $\mathcal{B}'$. Esto implica que $B \in \mathcal{B}'$ y luego $\mathcal{B} \subseteq \mathcal{B}'$. En particular, como todas las bases de una matroide $(S, \mathcal{I})$ tienen el mismo tamaño, se concluye que el cardinal común de los elementos de $\mathcal{B}'$ es k. Sea ahora $B \in \mathcal{B}'$. Como $B \in \mathcal{I}$ existe por definición $B' \in \mathcal{B}$ con $B \subseteq B'$. Pero $|B| = k = |B'|$ por lo que $B = B' \in \mathcal{B}$. Se concluye que $\mathcal{B}' \subseteq \mathcal{B}$ terminando la demostración.

Problema 2 Colaboración autorizada (8 puntos)

Sea $(S, \mathcal{C})$ un clutter que no contiene al vacío y que satisface el axioma de eliminación de circuitos (es decir, satisface las propiedades de la Sección 4.1 de [1]). Defina

$$\mathcal{I} = \{X \subseteq S \colon \forall C \in \mathcal{C}, C \not\subseteq X\}$$
$$\mathcal{C}' = \{X \in 2^S \setminus \mathcal{I} \colon \forall e \in X, X - e \in \mathcal{I}\}$$

Demuestre que $(S, \mathcal{I})$ es el sistema de independencia de una matroide (es decir, satisface cualquiera de las definiciones 2.1; 2.2 o 2.3 de [1]) y que su conjunto de circuitos $\mathcal{C}'$ satisface $\mathcal{C}' = \mathcal{C}$.

Solución:

Como el único subconjunto de $\emptyset$ es $\emptyset \notin \mathcal{C}$, tenemos que $\emptyset \in \mathcal{I}$. Por otro lado, si $X \subseteq Y \in \mathcal{I}$, entonces Y no contiene ningún $C \in \mathcal{C}$ y luego X tampoco, por lo que $X \in \mathcal{I}$. Esto prueba que $(S, \mathcal{I})$ es sistema de independencia.

Probemos aumento débil. Sea X, $Y \in \mathcal{I}$ con $Y \setminus X = \{a, b\}$ $X \setminus Y = \{c\}$. Supongamos por contradicción que existen $C, D \in \mathcal{C}$ con $C \subseteq X + a$ y $D \subseteq X + b$. Como $X \in \mathcal{I}$, debemos tener que $a \in C$ y $b \in D$, por lo que $C \neq D$. Por otro lado, como $Y \in \mathcal{I}$ debemos tener $C \not\subseteq Y = X + a + b - c$ y luego $c \in C$. Del mismo modo $D \not\subseteq Y = X + a + b - c$ y luego $c \in D$. Concluimos que $c \in C \cap D$. Usando eliminación de circuitos, existe $C' \in \mathcal{C}$ con $C' \subseteq C \cup D - c \subseteq X + a + b - c = Y$. Esto contradice el hecho que $Y \in \mathcal{I}$. Por lo que no pueden existir simultáneamente $C \subseteq X + a$ y $D \subseteq X + b$ con $C, D \in \mathcal{C}$. Se concluye que $X + a \in \mathcal{I}$ o $X + b \in \mathcal{I}$.

Para finalizar, sea $\mathcal{C}'$ el conjunto de circuitos de $(S, \mathcal{I})$. Si $C \in \mathcal{C}'$ entonces como C no está en $\mathcal{I}$, debe existir $C' \in \mathcal{C}$ con $C' \subseteq C$. Veamos que $C' = C$. En efecto si existiera $e \in C$ con $C' \subseteq C - e$, entonces tendríamos $C - e \notin \mathcal{I}$. Pero entonces C no satisface las condiciones para estar en $\mathcal{C}'$ lo cual es una contradicción. Se concluye que $C' = C$ y luego $C \in \mathcal{C}$. Con esto hemos probado que $\mathcal{C}' \subseteq \mathcal{C}$.

Tomemos ahora $D \in \mathcal{C}$. Por definición, $D \notin \mathcal{I}$. Por otro lado, si existiera $e \in D$ con $D - e \notin \mathcal{I}$ entonces habría un elemento $D' \in \mathcal{C}$ con $D' \subseteq D - e$. Esto contradice el hecho que $\mathcal{C}$ es un clutter. Sigue que para todo $e \in D$, $D - e \in \mathcal{I}$, es decir, $D \in \mathcal{C}'$. Con esto hemos probado que $\mathcal{C} \subseteq \mathcal{C}'$ y por el párrafo anterior concluimos que $\mathcal{C} = \mathcal{C}'$.

Problema 3 Colaboración autorizada (8 puntos)

Sea $r\colon 2^S \to \mathbb{N}$ una función que satisface las propiedades de normalización, aumentos unitarios y submodularidad débil (es decir, satisface las propiedades de la Sección 5.2 de [1]) Defina

$$\begin{aligned} \mathcal{I} &= \{X \subseteq S \colon r(X) = |X|\} \\ r'&\colon 2^S \to \mathbb{N} \\ r'(X) &= \max\{|I| \colon I \in \mathcal{I}, I \subseteq X\} \end{aligned}$$

Demuestre que $(S, \mathcal{I})$ es el sistema de independencia de una matroide (es decir, satisface cualquiera de las definiciones 2.1; 2.2 o 2.3 de [1]) y que su función de rango r' satisface $r' = r$.

Solución: Demostremos primero que para todo $A \subseteq B \subseteq S$,

$$r(B) - r(A) \in \{0, 1, 2, \ldots, |B \setminus A|\}. \tag{1}$$

En partícular, que r es creciente. Llamemos $B \setminus A = \{e_1, \ldots, e_k\}$ y $E_i = \{e_1, \ldots, e_i\}$ para $i \in [k] + 0$, entonces:

$$r(B) - r(A) = r(A \cup E_k) - r(A \cup E_0) = \sum_{i=1}^{k} r(A \cup E_i) - r(A \cup E_{i-1}) = \sum_{i=1}^{k} r(A \cup E_{i-1} + e_i) - r(A \cup E_{i-1})$$

Como cada sumando está en $\{0, 1\}$ debido a aumentos unitarios, se deduce que $r(B) - r(A) \in \{0, 1, \ldots, k\}$ con $k = |B \setminus A|$. De lo anterior usando $A = \emptyset$ se concluye para todo $B \subseteq S$,

$$0 \leq r(B) \leq |B|. \tag{2}$$

Probemos que $(S, \mathcal{I})$ es sistema de independencia. Como $r(\emptyset) = 0 = |\emptyset|$ tenemos que $\emptyset \in \mathcal{I}$. Por otro lado, sea $X \subseteq Y \in \mathcal{I}$. Sabemos entonces que $r(X) \leq |X|$ y que $r(Y) = |Y|$. Usando que $r(Y) - r(X) \leq |Y \setminus X|$ tenemos que $r(X) \geq r(Y) - |Y \setminus X| = |Y| - |Y \setminus X| = |X|$, y luego $r(X) = |X|$ y $X \in \mathcal{I}$. Luego $(S, \mathcal{I})$ es sistema de independencia.

Demostremos ahora aumento débil. Sean $X, Y \in \mathcal{I}$ con $Y \setminus X = \{a, b\}$, $X \setminus Y = \{c\}$. Supongamos que $X + a$, $X + b$ están ambos fuera de $\mathcal{I}$, luego usando (2) se tiene que para todo $e \in \{a, b\}$, $r(X + e) < |X + e| = |X| + 1 = r(X) + 1$. Por aumentos unitarios, concluimos que $r(X + e) = r(X)$. Pero entonces, $r(X) = r(X + a) = r(X + b)$ y submodularidad débil implica que $r(X) = r(X + a + b) = r(Y + c)$. De aquí, usando X, $Y \in \mathcal{I}$ y que r es creciente, tenemos $|X| = r(X) = r(Y + c) \geq r(Y) = |Y| = |X| + 1$ lo que es una contradicción. Por lo tanto $(S, \mathcal{I})$ es matroide.

Para finalizar, sea r' la función de rango de la matroide, sea $Y \subseteq S$. Notemos primero que si $Y \in \mathcal{I}$ entonces $r(Y) = |Y| = r'(Y)$. Por otro lado si $Y \notin \mathcal{I}$ pero existe $a \in Y$ tal que $Y - a \in \mathcal{I}$ entonces $r'(Y) = \max\{|I| \colon I \in \mathcal{I}, I \subseteq Y\} = |Y - a| = |Y| - 1$. Por otro lado, como $Y - a \in \mathcal{I}$, $r(Y - a) = |Y - a| = |Y| - 1 \geq r(Y)$, donde la última desigualdad viene del hecho que $Y \notin \mathcal{I}$. Como r es creciente, tenemos que $r(Y) = r(Y - a) = |Y| - 1$. Con esto, ya sabemos que $r(Y) = r'(Y)$ si Y es independiente o si Y contiene un independiente de tamaño $|Y| - 1$. Usemos esto, para probar que $r(Y) = r'(Y)$ para todo Y, por inducción en $k = |Y|$.

Si $k \in \{0, 1\}$ estamos listos por lo que ya probamos, entonces supongamos que $k \geq 2$ y que no existe un independiente de tamaño $|Y| - 1$ dentro de Y. Sea X una base de Y, sean a, b dos elementos distintos de $Y \setminus X$ y sea $Z = Y - a - b$, de modo que $Y = Z + a + b$. Como X es base de todos, $Z, Z + a, Z + b$ y $Z + a + b$, tenemos que $r'(Z) = r'(Z + a) = r'(Z + b) = r'(Z + a + b) = r'(Y)$. Usando hipótesis de inducción (como $Z + a, Z + b, Z$ tienen menos elementos que Y) tenemos también que $r(Z + a) = r'(Z + a)$, $r(Z + b) = r'(Z + b)$, $r(Z) = r'(Z)$. En particular, $r(Z) = r(Z + a) = r(Z + b)$ y por submodularidad débil tenemos que $r(Z + a + b) = r(Z)$. De esto deducimos que $r(Y) = r(Z + a + b) = r(Z) = r'(Z) = r'(Y)$.

Problema 4: (10 puntos)

En clases se verá una variante de BFS (o DFS) para encontrar los vértices alcanzables desde un vértice u en un grafo **dirigido** $G = (V, E)$. Puede usar dicho algoritmo o alguna modificación en este ejercicio.

Sea G un grafo no dirigido cuyas aristas han sido particionadas en s colores: $E_1 \cup E_2 \cup \cdots \cup E_s = E$. (Puede suponer que dado e su color es $c(e) = j \in [s]$). Un paseo alternante de u a v es un paseo de u a v cuyas aristas $e_1 e_2 \ldots e_k$ son tales que dos aristas consecutivas no son del mismo color.

(a) 3 puntos: Demuestre que si $s = 2$ y G es bipartito entonces el paseo alternante más corto de u a v (si existe) es un camino.

Solución: Sea W un paseo alternante más corto de u a v. Y supongamos que algún vértice w se repite, en particular tomemos dos apariciones de w en W y particionemos el paseo como $W = uW_1wW_2wW_3v$, es decir W_1 es un paseo de u a w, W_2 es un paseo (con al menos 2 aristas) de w a w y W_3 es un paseo de w a v. Sea L, R la bipartición de los vértices de V que hace a G bipartito y sin pérdida de generalidad suponemos que $w \in L$. Como W_2 es paseo que parte y termina en L y cada arista de un vértice de L debe llegar a un vértice de R, W_2 debe tener largo par. Como $s = 2$ y W_2 es alternante, sus aristas alternan de color en $\{1, 2\}$. Como W_2 tiene largo par, la última arista de W_2 tiene color distinto de la primera arista de W_2. Gracias a esto el paseo uW_1ww_3v obtenido al eliminar todas las aristas de W_2 también es alternante, y es más corto que W. Se concluye que no pueden haber vértices repetidos en el paseo alternante más corto.

(b) 3 puntos: Demuestre que para todo $s \geq 2$ y G cualquiera (no necesariamente bipartito), el paseo alternante más corto de u a v pasa por cada vértice a lo más 2 veces.

Solución:

Sea W el paseo alternante más corto de u a v, y supongamos que w aparece 3 o más veces en W. Descompongamos W como $uW_1wW_2wW_3wW_4v$ (donde W_2 y W_3 tienen al menos largo 2). Notamos que si $u = w$ entonces W_4 es u-v paseo alternante más corto que W y si $v = w$ entonces W_1 es u-v paseo alternante más corto que W, lo que no puede ser. Como $u \neq w$, W_1 tiene al menos una arista y como $w \neq v$, W_4 tiene al menos una arista.

Sea f la última arista de W_1 y e_2, e_3, e_4 las primeras arista de W_2, W_3, W_4 respectivamente. Analicemos sus colores. Si $c(e_3) = c(e_4)$ podemos eliminar W_3 para tener un paseo alternante más corto. Deducimos entonces que $c(e_3) \neq c(e_4)$. Luego, alguno de los dos, $c(e_3)$ o $c(e_4)$ es distinto a $c(f)$. Si $c(e_3) \neq c(f)$ entonces podemos eliminar W_2 para tener un paseo alternante más corto que W, y si $c(e_4) \neq c(f)$ entonces podemos eliminar todo W_2 y W_3 para tener un paseo alternante más corto que W. Concluimos que en cualquier caso, w no puede aparecer 3 veces.

(c) 4 puntos: (no relacionado con las partes anteriores). Sea G un grafo conexo cualquiera, $s \geq 2$, y $u \in V(G)$. Diseñe un algoritmo de complejidad $O((n + m)s)$ que encuentre el conjunto W de todos los vértices v para los cuales existe un paseo alternante de u a v en el grafo.

Solución: Sea u fijo. Primero notamos que basta usar paseos alternantes que partan en u y no usen a u nuevamente en su lista de vértices. Usaremos un grafo auxiliar para resolver el problema. Definamos $W = ((V - u) \times [s]) + u$, $F = \{(x, i)(y, j) \colon xy \in E, i \neq j, c(vw) = j\}$, $F' = \{u(x, j) \colon ux \in E, c(ux) = j\}$, y consideremos el grafo dirigido $H = (W, F \cup F')$. Notemos que cualquier paseo alternante Q en G que parta en u se puede codificar de manera como un paseo $\varphi(Q)$ en H que parte de u.

En efecto si escribimos alternadamente aristas y vértices de un paseo Q en G como $ue_1v_1e_2v_2\dots e_kv_k$, podemos encontrar la secuencia de vértices $\varphi(Q) = u(v_1, c(e_1))(v_2, c(e_2))\dots(v_k, c(e_k))$ que es un paseo en H por definición de F y F'. Similarmente, cualquier paseo en H que parta en u, $u(v_1, c_1)(v_2, c_2)\dots(v_k, c_k)$ corresponde a la secuencia de vértices $uv_1v_2\dots v_k$ en G, que es paseo por la manera como definimos F y F'. Más aún es alternante pues (si llamamos $v_0 = u$) para todo $i \in [k]$, $c(v_{i-1}v_i) = c_i$ y por definición de F, $c_i \neq c_{i+1}$. Esencialmente si un paseo en H pasa por un vértice (x, j), esto representa que en el grafo G se entró a x usando alguna una arista de color j. Con esto es claro que cualquier u-(x, j) paseo en G representa un u-x paseo alternante con última arista de color j.

Consideremos entonces el siguiente algoritmo: (1) Construir H. (2) Usar BFS en H para encontrar los vértices alcanzables X de H. (3) Devolver el conjunto $Y = \{u\} \cup \{x \in V - u\colon \exists j \in [m], (x, j) \in X\}$.

Este algoritmo entrega los alcanzables desde u via paseos alternantes (su correctitud es directa debido a la interpretación dada en los párrafos anteriores). El paso (1) solo requiere que podamos construir la lista de arcos que salen de cualquier nodo (x, i) (basta ciclar por todo $xy \in \delta^+_G(x)$, y en cada paso agregar $(x, i)(y, c(xy))$ si $c(xy) \neq i$). Es claro que esto toma tiempo $O(\delta^+_H(x, i))$. Lo mismo sucede para la lista de arcos que salen de u. En particular, el paso (1) se puede hacer en tiempo $O(|V(H)| + |E(H)|)$.

El paso (2) también se puede hacer en tiempo $O(|V(H)| + |E(H)|)$. El paso (3) se puede hacer en tiempo $O(|X|) = O(|V(H)|)$. Para concluir notamos que $|V(H)| = ns$, $|F| = O(ms)$, $|F'| = \delta^+_G(u) = O(m)$, por lo que el algoritmo anterior corre en tiempo $O(|V(H)| + |F| + |F'|) = O((n + m)s)$.

Problema 5: (10 puntos)

(a) 3 puntos: Sean a, b, c tres vértices de un grafo G tal que $bc \in E$. Sean P un camino de a a b; y Q un camino de a a c. Demuestre que si P y Q tienen la misma paridad (ambos de largo par; o ambos de largo impar) entonces G tiene un ciclo de largo impar.

Solución: Consideremos P y Q como conjuntos de aristas. Primero veamos el caso $a = b$ (notamos que $b \neq c$ pues existe arista $bc = ac$). En este caso $P = \emptyset$ que es de largo par, y por lo tanto Q es de largo par y $Q + ac$ es un ciclo de largo impar. El caso $a = c$ es análogo. Así que consideremos de ahora en adelante que $a \neq b$, $a \neq c$ (y $b \neq c$)

Probemos por inducción en $k = (|P|+|Q|)/2$ (recordar que $|P|+|Q|$ es par) que existe un ciclo en $(V, P\cup Q\cup\{bc\})$. Como tanto P como Q tienen aristas, el caso base es $k = 1$. En este caso, $P = \{ab\}$, $Q = \{ac\}$ y luego abc es un ciclo impar. Supongamos entonces $k \geq 2$. Consideremos el paseo W que parte usando P luego continua con la arista bc y luego continúa con el reverso de Q que va de c hasta a. Listemos los vértices de W y busquemos la primera vez que un vértice se repite: a, (vértices de P), $b, c, \dots, x$ donde x es un vértice ya visitado. Notamos que x debe ser un vértice de P también (incluyendo sus extremos). Si $x = a$ entonces $P \cup Q \cup \{ac\}$ son las aristas de un ciclo (que debe ser impar). Así que supongamos que $x \neq a$, en este caso tomemos P' como el subcamino de P que va de a a x; P'' el subcamino de P de x a b; Q' como el subcamino de Q que va de a a x, Q'' el subcamino que va de x a c. Llamemos y al penúltimo vértice de Q' y R al subcamino de Q' eliminando su última arista (y es extremo de R).

Notamos que P' es camino de a a x, R es camino de a a y, xy es arista. Si P' y R tienen la misma paridad entonces, por hipótesis de inducción encontramos un ciclo impar. Asi que supongamos que P' y R tienen distinta paridad. Notamos que $C = P''\cup\{bc\}\cup Q''$ es un ciclo que pasa por a. Su largo (módulo 2) es $|P''|+|Q''|+1$ y como 1 es igual a $|P'|+|R|$ módulo 2, tenemos que el largo de C (módulo 2) es $|P''|+|Q''|+|P'|+|R| = |P|+|Q-xy| = |P|+|Q|-1 = 1$. Es decir C tiene largo impar.j

(b) 3 puntos: Demuestre que G es bipartito si y solo si G no tiene ciclos de largo impar.
Indicación: Para la dirección complicada use la parte anterior.

Solución: Supongamos que G es bipartito con clases L y R. Si $C = v_1v_2\dots v_{2k+1}$ es un ciclo de largo impar $2k+1$ entonces tiene el mismo número de vértices, y sus vértices alternan clases. Esto implica que todos los vértices v_i con i impar están en la misma clase, pero v_1 y v_{2k+1} son vecinos lo que es contradicción. Para la otra dirección supongamos que G no tiene ciclos de largo impar. Probemos que cada componente conexa es bipartita. Consideremos entonces un vértice cualquiera a de una componente $K \subseteq V$ y definamos $L = \{x \in K\colon d(a, x) \text{ es par}\}$ donde $d(a, x)$ es el largo del camino más corto de a a x. Definamos $R = K \setminus L = \{x \in K\colon d(a, x) \text{ es impar}\}$. No puede haber una arista entre dos vértices b, c de L, pues si lo hubiera contradiría lo probado en la parte (a). Del mismo modo no hay vértices b, c de R con $bc \in E$. Se concluye que G es bipartito.

(c) 4 puntos: Sea G un grafo *no necesariamente conexo*. Diseñe un algoritmo de complejidad $O(n + m)$ para determinar si G es bipartito o no, y que en el caso que G sea bipartito reporte la bipartición L, Rs.
Indicación: Debe demostrar correctitud y esbozar la complejidad.

Solución: Si G fuera conexo, basta hacer BFS desde un vértice arbitrario a para determinar las distancias y los conjuntos L y R de la parte anterior. Luego podemos testear cada arista y ver si conecta dos vértices de la misma clase (el testeo se podría hacer durante el BFS pero es más claro si se hace al final). Si encontramos una entonces el grafo no es bipartito. Si no, entonces L y R son los conjuntos pedidos. La correctitud está dada por la parte anterior y la complejidad es $O(n + m)$ por el BFS y $O(m)$ por el testeo, en total $O(n + m)$

Para grafos generales basta primero encontrar las componentes conexas (en $O(n+m)$ usando variantes de BFS) y luego ejecutar el algoritmo anterior en cada una de ellas dando un total de $O(n + m)$. Es importante notar que no se puede agregar una vértice artificial v_0 y aristas hacia todos los vértices para hacer ambos pasos en uno solo, pues esto puede crear ciclos de largo impar.

Problema 6: (16 puntos)

El objetivo de este problema es demostrar correctitud y estudiar la complejidad del algoritmo de Borůvka visto en clases para encontrar un árbol generador de peso mínimo en un grafo conexo. De cátedra ya sabemos que si el grafo entregado al final (V, F) es acícliclo entonces es un MST, por lo que para correctitud solo debemos probar que el grafo entregado es acícliclo. Para simplificar la exposición, suponga que G es conexo y que $c\colon E \to \mathbb{R}$ es una función de costos **inyectiva**.

Llame $G_i = (V, F_i)$ al grafo al principio de la iteración i y sea $\mathcal{U}_I$ el conjunto de componentes conexas de G_i. En particular $\mathcal{U}_1 = \{\{v\}\colon v \in V\}$). El algoritmo elige para cada componente $U \in \mathcal{U}_i$ una arista $e_U = uv$ (la de menor peso en $\delta(U)$, que es única por inyectividad). Llame $f(e_U) \in \mathcal{U}_i$ a la componente a la cual pertenece el extremo de e_U que está fuera de U. Defina entonces $G'_i = (\mathcal{U}_i, F'_i)$ como el multigrafo sin loops cuyos vértices son las componentes en $\mathcal{U}_i$ y donde para cada $U \in \mathcal{U}_i$ se agrega una arista $\tilde{U} := Uf(e_U)$ a F'_i. Por simplicidad extienda la función de costos como: $c(\tilde{U}) = c(e_U)$.

(a) 3 puntos: Demuestre que los únicos ciclos del multigrafo G'_i tienen largo 2 (es decir son pares aristas paralelas). Más aún demuestre que si $\tilde{U}$ es una arista paralela a $\tilde{W}$ en H_i entonces $e_U = e_W$. **Indicación:** Parta suponiendo que $\tilde{U}$ es la arista más liviana de un ciclo.

Solución: Supongamos que C es un ciclo en G'_i de largo $k \geq 3$. Para hacer la exposición más simple dirijamos cada arista $\tilde{W} = Wf(e_W) \in C$ desde W hasta $f(e_U)$. En este ciclo con direcciones cada vértice W tiene grado de salida a lo más 1 (y por definición este arco que sale apunta a $f(e_W)$). De hecho es exactamente 1 pues C tiene tantos arcos como nodos y luego si W es vértice de C, $\tilde{W}$ debe ser arista de C. Ordenemos los vértices de C como $W_1, \dots, W_k$ de modo que $\tilde{W}_1 = W_1W_2, \tilde{W}_2 = W_2W_3, \dots, \tilde{W}_k = W_kW_1$ son sus aristas y sin pérdida de generalidad (reindexando) pensemos que $\tilde{W}_1 = W_1W_2$ es la arista más liviana de C. Pero tanto e_{W_1} como e_{W_2} son incidentes a W_2, y son aristas distintas pues e_{W_2} tiene un extremo en W_3, mientras que e_{W_1} tiene sus extremos en W_1 y W_2 (y supusimos $k \geq 3$). Como los costos son inyectivos y e_{W_2} es la arista más liviana de $\delta(W_2)$ concluimos que $c(W_1W_2) = c(e_{W_1}) > c(e_{W_2}) = c(W_2W_3)$, lo cual contradice nuestra elección de arista más liviana. Se concluye que k no puede ser mayor o igual que 3.

(b) 3 puntos: Sea $G''_i = (\mathcal{U}_i, F''_i)$ el grafo simple obtenido de G'_i al mantener solo una arista de cada par de aristas paralelas de G'_i. De la parte anterior se obtiene que G''_i no tiene ciclos. Use esto para demostrar por inducción que $G_i = (V, F_i)$ no tiene ciclos y concluya que el algoritmo entrega un árbol generador al final.

Indicación: Demuestre que si (V, F_i) es acíclico pero (V, F_{i+1}) tiene un ciclo C entonces se puede usar C para encontrar un ciclo en G''_i.

Solución: Observamos primero que $G_1 = (V, F_1)$ no tiene aristas y luego no tiene ciclos. Supongamos ahora que $i \geq 1$, y que G_i es acíclico. Sea C un ciclo de G_{i+1} con aristas $e_1, e_2, \dots, e_k$ (en ese orden). Llamemos C' a las aristas de C que fueron agregadas a F en la iteración i, es decir $C' = C \cap F_{i+1} \setminus F_i$. En otras palabras estas son las aristas de C con extremos en distintas partes de $\mathcal{U}_i$. Cada arista $e \in C'$ tiene una arista $\tilde{e}$ asociada en G_i'. Pero entonces las aristas $\{\tilde{e}\colon e \in C'\}$ forman un paseo cerrado en G_i' y luego contienen un ciclo en G_i', lo que solo puede ocurrir si este ciclo tiene largo 2 digamos $\tilde{e}\tilde{f}$. Pero entonces $\tilde{e}$ es una arista que conecta 2 componentes U y W (lo mismo $\tilde{f}$) y luego $\{e, f\} = \{e_U, e_W\}$. Digamos $e = e_U$, $f = e_W$, con $e \neq f$ Pero como ambas aristas están en $\delta(U)$ y ambas en $\delta(W)$ se tiene que $c(e_U) \leq c(e_W) \leq c(e_U)$ lo que implica que tienen igual costo y luego, por inyectividad de c, $e_U = e_W$, contradiciendo $e \neq f$. Se concluye que C no puede existir y luego G_{i+1} es acíclico.

(c) 7 puntos: Necesitamos dar una implementación más precisa para cada iteración del algoritmo. Para esto supongamos sin pérdida de generalidad que $V = [n]$.

Defina para cada vértice u el valor $f_i(u) = \text{mín}\{v\colon v \text{ está en la misma componente que } u \text{ en } G_i\}$.

Describa un algoritmo que dado (V, F_i) calcule $f_i(u)$ para cada $u \in V$ en tiempo $O(n + |F_i|)$.

Describa un segundo algoritmo que dado E, F_i, la función de costo c y todos los $f_i(u)$ calcule el conjunto de aristas $\{e_U : U \in \mathcal{U}_i\}$ a agregar en la iteración i-ésima en tiempo $O(n + m) = O(m)$

Use ambos algoritmos para concluir que (V, F_{i+1}) se puede calcular a partir de (V, F_i) en tiempo $O(m)$.

Solución, primer algoritmo Un algoritmo sin tanto nivel de detalle es (basta con esto para tenerlo completo)
(1) Crear un grafo H_i a partir de (V, F_i) agregando un vértice ficticio 0 y las aristas $0v$ para cada $v \in [n]$
(2) Encontrar un DFS T a partir de 0, pero de modo que primero se procesen las aristas $0i$ con menor índice (es decir, en el primer paso incorporar a la pila las aristas $0n$, $0(n-1)$, $\ldots$, 01 en ese orden, de modo que primero salga 01, luego 02, etc.)
(3) Sabemos que para cada vecino v de v_0 de T, el subárbol que cuelga de v en T genera la componente conexa de v. Por el paso anterior, el vértice v también es el de menor índice. Luego podemos volver a recorrer T usando DFS y marcando todos los descendientes w de $v \in N_T(v_0)$ con $f(w) = v$.

La complejidad del algoritmo es $O(n+|F_i|)$ para cada uno de los 3 pasos, resultando en un algoritmo $O(n+|F_i|)$. Alternativamente se puede hacer todo en un solo DFS como sigue:

Entrada: (V, F_i)
$U \leftarrow \emptyset$, $F \leftarrow \emptyset$
PILA $\leftarrow (0n, \ldots, 01)$.
AUX $\leftarrow 0$.
mientras *PILA* $\neq \emptyset$. **hacer**
  si $e = 0j$ **entonces**
    si $j \notin U$ **entonces**
      AUX $= j$
      $U \leftarrow U + j$ Insertar aristas de $\delta_{F_i}(j)$ en PILA
      $f_i(j) \leftarrow j$
    fin
  si no, si $e \in \delta(U)$, $u \in U$, $v \notin U$ **entonces**
    $U \leftarrow U + v$
    $F \leftarrow F + e$
    Insertar aristas de $\delta(v)$ en PILA
    $f_i(v) \leftarrow$ AUX
  fin
fin
devolver f_i

Solución, segundo algoritmo Para el segundo algoritmo, basta primero crear una lista $X(j)$ para cada j que sea mínimo de alguna componente U, que contenga las aristas de $\delta_{F_i}(U)$ y luego calcular el máximo en cada lista y añadirla al conjunto de aristas que se desea (basta con esto para tener completo). Formalmente esto se podría hacer en tiempo $O(n+m)$ como sigue:

Entrada: $(V, F_i), f_i$
Para cada $j \in [n]$, $X(j) \leftarrow \emptyset$.
Inicializar arreglo indicatriz F' con $F'(e) = 0$ para cada $e \in E$.
para $e = uv \in E$ **hacer**
  si $f_i(u) \neq f_i(v)$ **entonces**
    Agregar e a $X(f_i(u))$
    Agregar e a $X(f_i(v))$.
  fin
fin
para $j \in [n]$ **hacer**
  si $X(j) \neq \emptyset$ **entonces**
    Encontrar $e \in X(j)$ con $c(e)$ mínimo.
    $F'(e) \leftarrow 1$.
  fin
fin
devolver F'

La inicialización es $O(n+m)$, el primer loop es $O(m)$ y el segundo es $O(n)$ más el número total de elementos incorporados a algún $X(j)$. Como cada arista entra en a lo más dos $X(j)$, esto es $O(m)$. En total es $O(n+m)$.

Solución, conclusión
Usando que G es conexo, $n = O(m)$. Dado (V, F_i) podemos calcular f_i usando el primer algoritmo en tiempo $O(n + |F_i|) = O(n+m) = O(m)$. Luego podemos usar (V, F_i) y f_i mediante el segundo algoritmo para construir las aristas a agregar F' en tiempo $O(m)$. Finalmente F_{i+1} se obtiene agregando F' a F_i en tiempo $O(m)$.

(d) 3 puntos: Demuestre que el número de iteraciones del algoritmo es $O(\log n)$ y concluya de las partes anteriores que el algoritmo de Borůvka se puede implementar en tiempo $O((n+m)\log n) = O(m \log n)$.

Solución:
Notamos que si U es una componente de F_i, entonces al final de la i-ésima iteración la componente que contiene a U aumenta de tamaño (se fusiona con al menos otra componente, $f(U)$). En particular el número de componentes de F_{i+1} es a lo más la mitad del número de componentes de F_i. Como en F_1 hay n componentes, se necesitan a lo más $\log_2(n) + 1 = O(\log n)$ iteraciones para que el número de componentes sea 1 (y luego Borůvka termina en $O(\log n)$ iteraciones). Como cada iteración toma $O(n+m)$, tenemos que el algoritmo tiene complejidad total $O((n+m)\log n)$.

Referencias

[1] Soto, J.A. *Criptomorfismos en Matroides*, Material curso MA3705, `https://www.overleaf.com/read/tsnrcqwpbjtd`