Optimización Combinatoria y Teoría de Grafos

Docentes: Graciela Nasini, Daniel Severin, Pablo Torres

Práctica nº 3

1. El algoritmo de Kruskal se utiliza para hallar un *árbol generador de costo mínimo* de un grafo.

   Dado $G = (V, E)$ y $c \in \mathbb{Z}^E$, el algoritmo comienza inicializando $H = (V, F)$ con $F = \emptyset$ e indexando los arcos de G de forma que $E = \{e_1, e_2, \ldots, e_m\}$ verifique $c_{e_i} \leq c_{e_{i+1}}$ para todo $i = 1\ldots, m-1$. Luego, para cada $i = 1, \ldots, m$, hace:

   Si los extremos de e_i están en distintas componentes conexas de H, entonces $F \leftarrow F \cup \{e_i\}$.

   Utilice el algoritmo de Kruskal para encontrar un árbol generador mínimo de la instancia $G = (V, E); c$, con $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 6), (3, 4), (3, 5), (4, 5), (4, 6), (5, 6)\}$ y $c = (8, 18, 20, 23, 12, 30, 15, 30, 3, 7, 9)$. Compruebe sus iteraciones con la rutina de Scilab `http://www.fceia.unr.edu.ar/~daniel/OC/kruskal.sci`

   Ayuda para ejecutar `kruskal`:

   ```
   exec('kruskal.sci',-1)
   G(1,:) = [1 1 1 1 2 2 3 3 4 4 5];
   G(2,:) = [2 3 4 5 3 6 4 5 5 6 6];
   costos = [8 18 20 23 12 30 15 30 3 7 9];
   [F, valor] = kruskal(G, costos)
   ```

2. Sea $G = (V, E)$ un grafo.

   a) Pruebe que, si $\mathcal{S} = E$ e $\mathcal{I} = \{J \subseteq \mathcal{S} : J$ es bosque de $G\}$, $(\mathcal{S}, \mathcal{I})$ es una matroide.

   b) Pruebe que si G es conexo, todo bosque maximal es un árbol.

   c) Pruebe que el *Problema del Árbol Generador Mínimo* se puede reducir al *Problema del Bosque de Peso Máximo*.

   d) Recordando que el algoritmo goloso resuelve correctamente el *Problema de Conjunto Independiente de Costo Máximo* sobre toda matroide y utilizando los items anteriores, pruebe que el algoritmo de Kruskal resuelve correctamente el *Problema del Árbol Generador Mínimo*.

3. Determine el orden de complejidad del algoritmo de Kruskal sobre una instancia $G = (V, E)$ con $|V| = n$, $|E| = m$, y $c \in \mathbb{R}^E$. Recuerde que el orden de complejidad está asociado al peor caso y tenga en cuenta las observaciones siguientes.

   Observaciones: La mejor implementación de Kruskal utiliza una representación de los datos que permite realizar cada operación con la siguiente complejidad:

Descripción de la operación	Cant. oper. elementales
1) Ordenar los arcos de E según su costo	$C_1 m \log m$
2) Inicializar $H = (V, F)$ con F vacío	$C_2 n$
3) Verificar si dos vértices dados pertenecen a la misma componente en H	C_3
4) Agregar una arista (u, v) a F tal que $u \in V_1$ y $v \in V_2$ donde V_1 y V_2 son diferentes componentes conexas en H	$C_4 \min\{\lvert V_1\rvert, \lvert V_2\rvert\}$

donde C_1, C_2, C_3 y C_4 son constantes.

De esta manera, el peor caso para el algoritmo de Kruskal se daría cuando, en cada iteración en la que se agrega un arco, la componente conexa de H de menor cardinal es *lo más grande posible*. Observar que Kruskal agrega un arco en $n-1$ iteraciones de las m que realiza y que en la i-ésima iteración en que agrega un arco, el grafo H tiene $n-i+1$ componentes conexas.

- Pruebe que en un grafo con n vértices y k componentes conexas, la componente de menor cardinalidad tiene a lo sumo $\left\lfloor \frac{n}{k} \right\rfloor$ vértices.
- Para el orden de la complejidad del peor caso utilice la siguiente propiedad:

$$\sum_{k=1}^{n-1} \frac{1}{k} \leq 1 + ln(n-1).$$

4. Pruebe que las siguientes estructuras $(\mathcal{S}, \mathcal{I})$ son matroides:

   a) *Subconjuntos de columnas l.i.*: Sea $A \in \mathbb{R}^{m \times n}$, $\mathcal{S} = \{1, \ldots, n\}$ e $\mathcal{I} = \{J \subseteq \mathcal{S}$: las columnas indexadas por los elementos de J son l.i.$\}$.

   b) *Uniforme*: Sea $k \in \mathbb{N}$, $\mathcal{S}$ un conjunto finito e $\mathcal{I} = \{J \subseteq \mathcal{S} : |J| \leq k\}$.

5. Considere un grafo bipartito $G = (V_1 \cup V_2, E)$, es decir $E \subseteq V_1 \times V_2$.

   a) Pruebe que si $\mathcal{S} = E$ e $\mathcal{I}_M = \{J \subseteq E : J$ es un matching de $G\}$ entonces $(\mathcal{S}, \mathcal{I}_M)$ es un sistema independiente pero no un matroide.

   b) Sea $\mathcal{S} = E$. Pruebe que, para $i = 1, 2$, si

   $$\mathcal{I}_i = \{J \subseteq E : \text{cada } v \in V_i \text{ es incidente en a lo sumo un arco de } J\},$$

   $(\mathcal{S}, \mathcal{I}_i)$ es una matroide.

   c) Pruebe que $\mathcal{I}_M = \mathcal{I}_1 \cap \mathcal{I}_2$.

6. Sea un grafo $G = (V, E)$ y un conjunto $M \subseteq E$. Pruebe que M es un matching de G si y sólo si M es un estable de $L(G)$.

7. Se dice que una propiedad $\mathscr{P}$ sobre grafos es *hereditaria* (por subgrafos inducidos por nodos) si para cualquier grafo G que satisface $\mathscr{P}$, todo subgrafo inducido por nodos de G también satisface $\mathscr{P}$. Determine cuáles de las siguientes propiedades son hereditarias:

   a) G es bipartito

   b) G es planar

c) G es grafo de línea

d) G tiene un circuito euleriano

e) G tiene un circuito hamiltoniano

f) G tiene un k-coloreo

g) G tiene un matching perfecto.

8. En *Teoría de Grafos* es usual trabajar con familias definidas de la forma

$$\mathcal{G} = \{G : G \text{ es un grafo con la propiedad } \mathscr{P}\}.$$

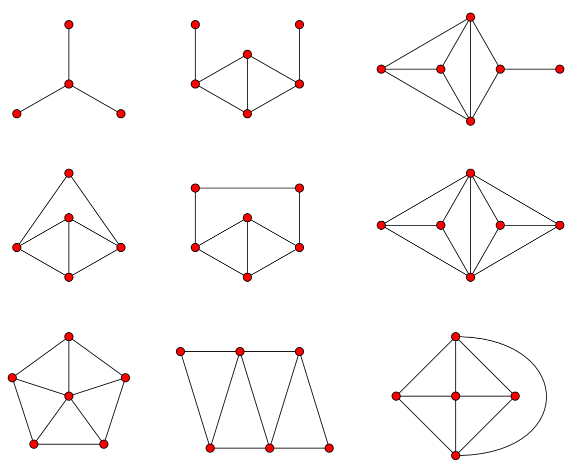
Si no se conocen algoritmos polinomiales que permitan decidir si un grafo cualquiera tiene o no dicha propiedad, se trabaja en la determinación de condiciones suficientes y de condiciones necesarias para que un grafo pertenezca a $\mathcal{G}$. Cuando $\mathscr{P}$ es una propiedad hereditaria, una de las líneas de abordaje es la identificación de *subgrafos prohibidos*.

Decimos que G es *minimamente no* $\mathscr{P}$ si $G \notin \mathcal{G}$ y, para todo $v \in V(G)$, $G \backslash \{v\} \in \mathcal{G}$.

a) Pruebe que $G \in \mathcal{G}$ si y sólo si G no contiene como subgrafo inducido ningún grafo mínimamente no $\mathscr{P}$.
Observación: Por esta propiedad, los grafos mínimamente no $\mathscr{P}$ se denominan *subgrafos prohibidos de* $\mathcal{G}$.

b) ¿Cuáles son los subgrafos prohibidos de los grafos bipartitos? Justifique.

9. Considere la siguiente lista de grafos:



a) Pruebe que, en todo grafo de línea, sus aristas pueden ser particionadas en subgrafos completos de modo tal que todo nodo pertenece a lo sumo a dos subgrafos completos.

b) Elija dos grafos de esta familia, uno de 5 vértices y otro de 6 y pruebe que son mínimamente no de línea.

Observación: Se sabe que los grafos de la lista son todos los subgrafos prohibidos de los grafos de línea [GC, pág. 110].

10. **Opcional:** Para las operaciones (2), (3) y (4) en las observaciones del Ejercicio 3, escriba un procedimiento en términos de operaciones elementales que satisfaga las cantidades dadas en la tabla.
Sugerencia: Represente las componentes conexas del grafo $H = (V, F)$ mediante una matríz $A = (a_{ij}) \in \mathbb{R}^{3\times n}$ de la siguiente forma:

- a_{1v} indica el primer elemento de la componente en que se encuentra v,
- a_{2v} indica la cantidad de elementos de dicha componente en caso de que v sea el primero de ella, o un valor indefinido $(-)$ en caso contrario,
- a_{3v} indica el *siguiente* elemento en la componente, o cero en caso que v sea el último.

Por ejemplo, una representación posible para la partición de los vértices $\{1,\ldots,8\}$ en las componentes conexas $\{1,6,3\}$, $\{5,2\}$, $\{4,7\}$, $\{8\}$ es la siguiente:

$$A = \begin{pmatrix} 1 & 5 & 1 & 4 & 5 & 1 & 4 & 8 \\ 3 & - & - & 2 & 2 & - & - & 1 \\ 6 & 0 & 0 & 7 & 2 & 3 & 0 & 0 \end{pmatrix}$$

Entonces, por ejemplo en la operación (3), para saber si los vértices 3 y 5 pertenecen a diferentes componentes, basta con verificar que $a_{13} \neq a_{15}$.

Bibliografía:

[LCO] W. Cook, W. Cunningham, W. Pulleyblank A. Schrijver. *Combinatorial Optimization.* Wiley-Interscience.

[GC] Andreas Brandstädt, Van Bang Le, Jeremy P. Spinrad. *Graph classes: a survey.* SIAM monographs on discrete mathematics and applications, 1999.