FACULTAD DE CIENCIAS EXACTAS INGENIERÍA Y AGRIMENSURA
Escuela de Posgrado y Educación Continua

Optimización Combinatoria y Teoría de Grafos

Docentes: Graciela Nasini, Daniel Severin

Práctica nº 2

1. Sea G un grafo r-regular (todos sus vértices tienen grado r) y k tal que $1 \leq k < r$. Pruebe que el problema de k-dominación se puede reducir polinomialmente al problema de $(r+1-k)$-empaquetamiento. Pruebe también que el problema de k-empaquetamiento se puede reducir polinomialmente al problema de $(r+1-k)$-dominación.

2. 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

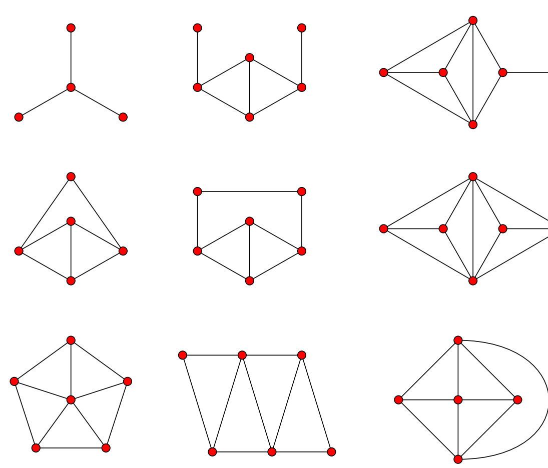
   e) G tiene un circuito hamiltoniano

   f) G tiene un k-coloreo

   g) G tiene un matching perfecto.

3. Sea $\mathscr{P}$ una propiedad hereditaria y $\mathcal{G}_{\mathscr{P}} = \{G : G \text{ es un grafo con la propiedad } \mathscr{P}\}$. Decimos que G es *minimamente no* $\mathscr{P}$ si $G \notin \mathcal{G}_{\mathscr{P}}$ y para todo $v \in V(G)$, $G \backslash \{v\} \in \mathcal{G}_{\mathscr{P}}$. Pruebe que $G \in \mathcal{G}_{\mathscr{P}}$ si y sólo si G no contiene como subgrafo inducido ningún grafo minimamente no $\mathscr{P}$.
   Comentario: Es común en la *Teoría de Grafos* caracterizar una familia $\mathcal{G}_{\mathscr{P}}$ a partir de una propiedad hereditaria $\mathscr{P}$ por *subgrafos prohibidos*. Esto significa hallar todos los subgrafos minimamente no $\mathscr{P}$. Por ejemplo, en el siguiente ejercicio se muestran todos los subgrafos prohibidos para caracterizar a los grafos de línea.

4. Pruebe que los siguiente grafos son minimamente no de línea.



5. 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 $L(G)$.

6. Dado un digrafo $D = (V, A)$, un *k-coloreo aditivo orientado de D* es una asignación $f : V \to \{1, \ldots, k\}$ tal que, para todo arco $(u, v) \in A$, la suma de las etiquetas de los vecinos de u es inferior a la suma de las etiquetas de los vecinos de v, es decir

$$\sum_{w \in N(u)} f(w) < \sum_{w \in N(v)} f(w).$$

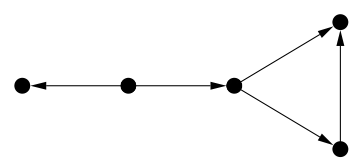
Nota: $N(v) = N^+(v) \cup N^-(v)$.

   a) Demuestre que si un digrafo D posee un ciclo dirigido, entonces no existe ningún k para el cual D tiene un k-coloreo aditivo orientado.

   b) De una condición necesaria y suficiente (en término de los grados de los vértices) para que un digrafo acíclico tenga un 1-coloreo aditivo orientado.

   c) Pruebe que un digrafo acíclico tal que su grafo subyacente es completo tiene un $|V|$-coloreo aditivo orientado.

   d) Demuestre que no existe ningún k para el cual el siguiente digrafo tiene un k-coloreo aditivo orientado.



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

   a) *Bosques*: Sea $G = (V, E)$ un grafo, $\mathcal{S} = E$ e $\mathcal{I} = \{J \subseteq \mathcal{S} : J$ es bosque de $G\}$.

   b) *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.$\}$.

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

8. 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) Pruebe que $(\mathcal{S}, \mathcal{I}_1)$ y $(\mathcal{S}, \mathcal{I}_2)$ son matroides, donde $\mathcal{S} = E$ y, para $i = 1, 2$,

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

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

9. Sea $D = (V, A)$ un digrafo. Decimos que D tiene dos vértices distinguidos $s, t \in V$ llamados *fuente* y *sumidero* respectivamente cuando $N^-(s) = N^+(t) = \emptyset$ y existe un st-camino dirigido en D. Sean $c_e \in \mathbb{Z}_+^A$ capacidades asociadas a cada arco. Decimos que $f \in \mathbb{Z}_+^A$ es un *st-flujo* de D si satisface la *restricción de capacidad* $f_e \leq c_e$ para todo $e \in A$ y la *conservación del flujo* $\sum_{v \in N^+(u)} f_{uv} = \sum_{v \in N^-(u)} f_{vu}$ para todo $u \in V \backslash \{s, t\}$. Denotamos con $val(f)$ al valor $\sum_{v \in N^+(s)} f_{sv} = \sum_{v \in N^-(t)} f_{vt}$. Considere el siguiente problema:

Problema de Máximo Flujo

Entrada: un digrafo $D = (V, A)$ con fuente s y sumidero t, y capacidades $c_e \in \mathbb{Z}_+^A$.
Objetivo: hallar un st-flujo f de D que maximiza $val(f)$.

Para una instancia dada de este problema, el algoritmo de *Ford-Fulkerson* la resuelve en tiempo polinomial [LCO, pág. 47]. Teniendo esto en cuenta, demuestre que el *Problema de Máximo Matching en Bipartitos* es polinomial también.
Sugerencia: Consulte [LCO, pág. 47].

Problema de Máximo Matching en Bipartitos

Entrada: un grafo bipartito $G = (V, E)$.
Objetivo: hallar un matching $M \subset E$ tal que $|M|$ es máximo.

10. El algoritmo de Kruskal, luego de ordenar los arcos de la forma $\{e_1, e_2, \ldots, e_m\}$ con $c_{e_i} \leq c_{e_{i+1}}$ para todo $i = 1 \ldots, m-1$ e inicializar $H = (V, F)$ con $F = \emptyset$, realiza los siguientes pasos:

    Para $i = 1, \ldots, m$ hacer:
        Si *los extremos de e_i estan en distintas componentes conexas de H*, hacer:
            $F \leftarrow F \cup \{e_i\}$

    a) Resuelva con este algoritmo la instancia del Ejercicio 14 de la Práctica 1.

    b) La complejidad computacional de este algoritmo depende de la manera en que se implemente la iteración i-ésima con $i = 1, \ldots, m$. La implementación dada por Kruskal utiliza una estructura de datos llamada *disjoint-set linked lists* que, dado el conjunto $V = \{1, \ldots, n\}$, permite *operar* con una partición de V que se representa mediante una matríz $A = (a_{ij}) \in \mathbb{R}^{3\times n}$: para un $v \in V$ dado, 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, y 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 $\{1, 6, 3\}$, $\{5, 2\}$, $\{4, 7\}$, $\{8\}$ es la siguiente:

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

    Determine la complejidad computacional de inicializar la matríz A de manera que cada elemento esté en una componente propia, es decir $\{1\}$, $\{2\}$, …, $\{n\}$.

    c) Determine la complejidad computacional de chequear si dos elementos dados $u, v \in V$ pertenecen a la misma componente o no.

    d) Para dos elementos dados $u, v \in V$ tales que no pertenecen a la misma componente, la operación que une las componentes donde están u y v se realiza con el siguiente procedimiento:

    Si $a_{2a_{1u}} < a_{2a_{1v}}$ entonces: (v_1 1er. vért. de la comp. más chica; v_2 el 1ero. de la otra)
        $v_1 \leftarrow a_{1u}$, $v_2 \leftarrow a_{1v}$
    sino
        $v_1 \leftarrow a_{1v}$, $v_2 \leftarrow a_{1u}$

$a_{2v_2} \leftarrow a_{2v_1} + a_{2v_2}$ (calculamos el cardinal de la nueva componente)
$r \leftarrow a_{3v_2}$ (r tiene el 2do. vért. de la 2da. comp.)
$a_{3v_2} \leftarrow v_1$ (luego del 1er. vért. de la 2da. comp. viene el 1ero. de la 1era.)
$w \leftarrow v_2$ (los vért. de la 1er. comp. van a pasar a ser de la 2da.)
Para todo $i = 1, \ldots, a_{2v_1}$ hacer $w \leftarrow a_{3w}$, $a_{1w} \leftarrow v_2$
$a_{3w} \leftarrow r$ (luego del último vért. de la 1er. comp. viene el 2do. de la 2da.)

Cuente la cantidad de operaciones elementales de este procedimiento.

e) Sea n una potencia de dos. Determine una función de n que cuenta la cantidad de operaciones elementales al realizar múltiples uniones de la siguiente forma: comience con una partición de 1 elemento en cada componente, luego aplique $n/2$ uniones (unir $\{1\}$ con $\{2\}$, $\{3\}$ con $\{4\}$, etc) hasta obtener 2 elementos en cada componente, luego aplique $n/4$ uniones (unir $\{1,2\}$ con $\{3,4\}$, etc) hasta obtener 4 elementos en cada componente, etc, y finalice al obtener una sola componente.
Sugerencia: Por inducción sobre p, donde $n = 2^p$.

f) En la implementación de Kruskal, al inicializar H también se inicializa una matríz A que representa una partición de los vértices en donde cada vértice pertenece a una componente conexa propia. Luego, en la iteración i-ésima se toma $e_i = (u, v)$ y se chequea si u y v son de diferentes componentes, y en tal caso (además de agregarse e_i a F) se aplica el procedimiento anterior para unir las componentes de u y v en A. Teniendo esto en cuenta y sabiendo que el ordenamiento inicial tiene complejidad $O(m\ log\ m)$, demuestre que esta implementación tiene complejidad $O(m\ log\ m)$.

Bibliografía:

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