



## DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

### Algoritmos y Estructuras de Datos III

Primer cuatrimestre 2021 (*dictado a distancia*)

Matching, conjunto independiente y recubrimientos

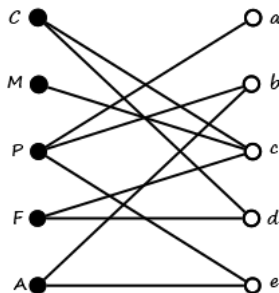
# Matching - Conjunto Independiente - Recubrimientos

Analicemos algunas situaciones:

- ▶ Hay un conjunto de cinco personas y un conjunto de 5 trabajos para realizar.
- ▶ Sean las personas Carlos, Marcela, Pedro, Fernando y Andrea, y los trabajos  $a$ ,  $b$ ,  $c$ ,  $d$  y  $e$ .
- ▶ Carlos está capacitado para realizar los trabajos  $c$  y  $d$ ,
- ▶ Marcela para  $c$ ,
- ▶ Pedro para  $a$ ,  $b$  y  $e$ ,
- ▶ Fernando para  $c$  y  $d$ ,
- ▶ y Andrea para  $b$  y  $e$ .
- ▶ ¿Es posible realizar una distribución del trabajo de modo que se puedan realizar todos los trabajos simultáneamente?

## Matching - Conjunto Independiente - Recubrimientos

Podemos modelar el problema mediante un grafo  $G = (V, X)$  donde el conjunto de vértices representa a las personas y los trabajos, y hacemos un vértice representando a una persona adyacente a los vértices correspondientes a los trabajos para los que está capacitado. Obtenemos el siguiente grafo:



## Matching - Conjunto Independiente - Recubrimientos

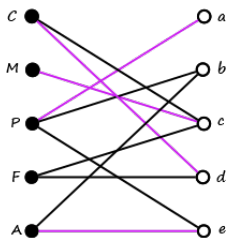
Para poder realizar todos los trabajo simultáneamente, debemos asignarle exáctamente un trabajo a cada persona. Intentemos:

- ▶ El trabajo  $a$  sólo puede realizarlo Pedro, así que no tenemos elección: le asignamos a Pedro el trabajo  $a$ .
- ▶ Marcela sólo sabe realizar el trabajo  $c$ , tampoco tenemos elección: le asignamos a Marcela el trabajo  $c$ .
- ▶ Carlos está capacitado para realizar los trabajos  $c$  y  $d$ , pero el  $c$  ya lo tiene asignado Marcela: le asignamos el trabajo  $d$  a Carlos.
- ▶ Fernando puede realizar los trabajos  $c$  y  $d$ , pero ambos trabajos ya están asignados, por lo que no tenemos trabajo para asignarle Fernando.

Esto ya muestra que no será posible realizar todos los trabajos simultáneamente.

## Matching - Conjunto Independiente - Recubrimientos

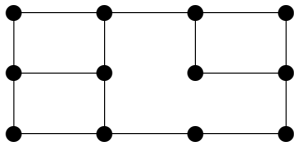
Andrea puede realizar los trabajos  $b$  o  $e$ . Ambos trabajos todavía están libres y le podemos asignar alguno de los dos, por ejemplo el trabajo  $e$ . Entonces sí podemos realizar simultáneamente 4 de los 5 trabajos.



Problema de asignación de personal: queremos encontrar un conjunto de aristas de cardinal máximo, tal que no haya dos aristas en el conjunto que incidan sobre el mismo vértice.

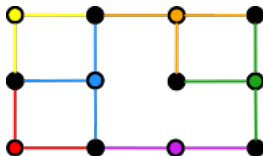
## Matching - Conjunto Independiente - Recubrimientos

El grafo de la figura representa el mapa de una ciudad. Se quiere ubicar policías en las esquinas de modo que todas las cuadras estén bajo vigilancia, o sea, cada cuadra tiene que tener un policía al menos en una de las esquinas. ¿Cuál es el mínimo número de policías necesarios?



## Matching - Conjunto Independiente - Recubrimientos

- Si ubicamos un policía en cada uno de los vértices coloreados, todas las cuadras estarán cubiertas.



- En este caso estamos necesitando 6 policías.
- Pero, ¿habrá forma de hacerlo con 5?
- Ahora nuestro objetivo es encontrar un conjunto de vértices de cardinal mínimo, tal que toda arista del grafo incida, al menos, sobre un vértice de este conjunto.

## Matching - Conjunto Independiente - Recubrimientos

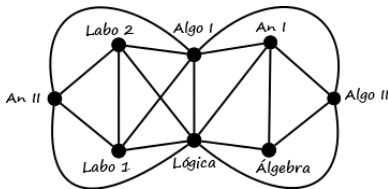
Volvamos al problema de las aulas de la clase de coloreo, pero con un objetivo distinto.

Supongamos que no tenemos restricción en cantidad de aulas y que el DC equipó una sala (sólo una por ahora) con la última tecnología y quiere que sea utilizada por la mayor cantidad de materias posible.

Volvamos a dibujar el grafo que representa nuestra instancia.



# Matching - Conjunto Independiente - Recubrimientos

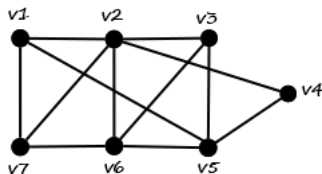


- ▶ Si le asignamos la sala a Lógica, ninguna otra materia podrá usarla, porque Lógica se solapa en horario con todas las demás.
- ▶ En cambio, si le asignamos la sala a Álgebra, también la podría utilizar por ejemplo Algo I (o Labo 1, o Labo 2, o Análisis II), pero no Algo II ni Análisis I ni Lógica. La sala sería utilizada por dos materias.
- ▶ Pero, ¿será posible que la sala sea utilizada por tres o más materias?
- ▶ Lo que queremos hallar es un conjunto de vértices de cardinal máximo tal que no haya dos vértices adyacentes dentro del conjunto.

# Matching - Conjunto Independiente - Recubrimientos

- ▶ Un *matching o correspondencia* entre los vértices de  $G$ , es un conjunto  $M \subseteq X$  de aristas de  $G$  tal que para todo  $v \in V$ ,  $v$  es incidente a lo sumo a una arista  $e \in M$ .
- ▶ Un *conjunto independiente* de vértices de  $G$ , es un conjunto de vértices  $I \subseteq V$  tal que para toda arista  $e \in X$ ,  $e$  es incidente a lo sumo a un vértice  $v \in I$ .
- ▶ Un *recubrimiento de las aristas* de  $G$ , es un conjunto  $R_n$  de vértices tal que para todo  $e \in X$ ,  $e$  es incidente al menos a un vértice  $v \in R_n$ .
- ▶ Un *recubrimiento de los vértices* de  $G$ , en este caso debemos pedir que  $G$  no tenga vertices aislados, es un conjunto  $R_e$  de aristas tal que para todo  $v \in V$ ,  $v$  es incidente al menos a una arista  $e \in R_e$ .

# Matching - Conjunto Independiente - Recubrimientos



$$M = \{(v_1, v_7), (v_2, v_6), (v_3, v_5)\} \quad I = \{v_1, v_6, v_4\}$$

$$R_a = \{v_1, v_2, v_5, v_6\} \quad R_v = \{(v_1, v_7), (v_2, v_6), (v_3, v_5), (v_4, v_5)\}$$

# Matching - Conjunto Independiente - Recubrimientos

Los problemas de optimización relacionados a estos conceptos son:

- ▶ El problema de *matching máximo* consiste en encontrar un matching de cardinal máximo entre todos los matchings de  $G$ .
- ▶ El problema de *conjunto independiente máximo* consiste en encontrar un conjunto independiente de cardinal máximo entre todos los conjuntos independientes de  $G$ .
- ▶ El problema de *recubrimiento de aristas mínimo* consiste en encontrar un recubrimiento de aristas de cardinal mínimo entre todos los recubrimientos de aristas de  $G$ .
- ▶ El problema de *recubrimiento de vértices mínimo* consiste en encontrar un recubrimiento de vértices de cardinal mínimo entre todos los recubrimientos de vértices de  $G$ .

# Matching - Conjunto Independiente - Recubrimientos

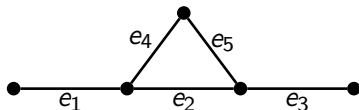
Nuevamente, aunque parezcan problemas muy parecidos, unos sobre vértices y otros sobre aristas, computacionalmente son muy distintos:

- ▶ Para el problema de matching máximo existen algoritmos polinomiales para resolverlo.
- ▶ Mientras que para conjunto independiente máximo no se conoce ninguno (y se piensa que no existe).

# Matching - Conjunto Independiente - Recubrimientos

**Lema:**  $S \subseteq V$ .  $S$  es un conjunto independiente  $\iff V \setminus S$  es un recubrimiento de aristas.

Esta relación no se mantiene entre matchings y recubrimientos de vértices, como lo muestra este ejemplo:



$$M = \{e_3, e_4\}$$

$$X \setminus M = \{e_1, e_2, e_5\}$$

no es recubrimiento

$$R = \{e_1, e_3, e_4\}$$

$$X \setminus R = \{e_2, e_5\}$$

no es matching

# Relación entre coloreo y conjunto independiente

- ▶ Ahora que conocemos la definición de conjunto independiente podemos definir el problema de coloreo de vértices desde un enfoque equivalente:
  - ▶ Cada conjunto de vértices pintados con el mismo color es un conjunto independiente.
  - ▶ Entonces, podemos definir un coloreo de  $G$  como una partición de sus vértices en conjuntos independientes.
  - ▶ El número cromático de  $G$ , es la cantidad de conjuntos de la partición con menos conjuntos.
- ▶ No es cierto que en un coloreo óptimo de  $G$  los vértices de un conjunto independiente de cardinal máximo estén siempre pintados todos del mismo color.
- ▶ Sin embargo, con esta idea podemos definir una heurística golosa para colorear un grafo.

## Relación entre coloreo y conjunto independiente

*coloreoPorCI*( $G$ )

**entrada:**  $G = (V, X)$

**salida:**  $f$  coloreo de los vertices de  $G$

$i \leftarrow 0$

**mientras**  $V \neq \emptyset$

$i \leftarrow i + 1$

$S \leftarrow \text{conjIndepMaximal}(G[V])$

**para** todo  $v \in S$  **hacer**

$f[v] \leftarrow i$

**fin para**

$V \leftarrow V \setminus S$

**retornar**  $f$



# Relación entre coloreo y conjunto independiente

- ▶ Notar que en cada iteración del ciclo,  $S$  es un conjunto independiente maximal, no necesariamente máximo.
- ▶ Encontrar un conjunto independiente maximal es fácil computacionalmente, mientras que encontrar uno máximo no lo es.
- ▶ Un procedimiento constructivo goloso para encontrar un conjunto independiente maximal *grande* de un grafo  $G$  podría incorporar en cada iteración el vértice candidato adyacente a la menor cantidad de vértices candidatos, donde un vértice es candidato si no es adyacente a ningún vértice ya incorporado al conjunto.

# Relación entre coloreo y conjunto independiente

*conjIndepMaximal*( $G$ )

**entrada:**  $G = (V, X)$

**salida:**  $I$  conjunto independiente maximal

$I \leftarrow \emptyset$

**mientras**  $V \neq \emptyset$  **hacer**

$v \leftarrow \arg \min \{|N(v) \cap V| : v \in V\}$

$I \leftarrow I \cup \{v\}$

$V \leftarrow V \setminus \{u : u \in V \text{ y } (u, v) \in X\}$

**retornar**  $S$

# Matching máximo

- ▶ Un vértice  $v$  se dice *saturado* por un matching  $M$  si hay una arista de  $M$  incidente a  $v$ .
- ▶ Dado un matching  $M$  en  $G$ , un *camino alternado* en  $G$  con respecto a  $M$ , es un camino simple donde se alternan aristas que están en  $M$  con aristas que no están en  $M$ .
- ▶ Dado un matching  $M$  en  $G$ , un *camino de aumento* en  $G$  con respecto a  $M$ , es un camino alternado entre vértices no saturados por  $M$ .

# Matching máximo

**Lema:** Sean  $M_0$  y  $M_1$  dos matching en  $G$  y sea  $G' = (V, X')$  con  $X' = (M_0 - M_1) \cup (M_1 - M_0)$ . Entonces las componentes conexas de  $G'$  son de alguno de los siguientes tipos:

- ▶ vértice aislado
- ▶ circuito simple con aristas alternadamente en  $M_0$  y  $M_1$
- ▶ camino simple con aristas alternadamente en  $M_0$  y  $M_1$ .

**Teorema:**  $M$  es un matching máximo de  $G$  si, y sólo si, no existe un camino de aumento en  $G$  con respecto a  $M$ .

## Matching máximo

Este teorema deriva en el siguiente algoritmo:

*matchingMaximo*( $G$ )

**entrada:**  $G = (V, X)$

**salida:**  $M$  matching maximo de  $G$

$M \leftarrow e$  una arista cualquiera

**mientras** exista camino de aumento en  $G$   
con respecto a  $M$  **hacer**

$P \leftarrow \text{caminoAumento}(G, M)$

$M \leftarrow M \oplus P$

**fin mientras**

**retornar**  $M$

**Teorema:** El algoritmo *matchingMaximo* aplicado a  $G$  retorna un matching de cardinal máximo de  $G$ .

# Matching máximo

- ▶ Nos queda pendiente tratar el problema de encontrar un camino de aumento de un grafo con respecto a un matching.
- ▶ Por su complejidad, lo dejaremos afuera de la materia, lo único que vamos a decir es que existen algoritmos polinomiales para resolverlo.
- ▶ Como el ciclo a lo sumo se realiza  $m$  veces y cada operación dentro del ciclo es polinomial, resulta un algoritmo polinomial para calcular un matching máximo de un grafo.

## Matching - Conjunto Independiente - Recubrimientos

**Teorema:** Dado un grafo  $G$  sin vértices aislados, si  $M$  es un matching máximo de  $G$  y  $R_e$  un recubrimiento mínimo de los vértices de  $G$ , entonces  $|M| + |R_e| = n$ .

**Teorema:** Dado un grafo  $G$ , si  $I$  es un conjunto independiente máximo de  $G$  y  $R_n$  un recubrimiento mínimo de las aristas de  $G$ , entonces  $|I| + |R_n| = n$ .

# Problema del cartero chino

Dado un grafo  $G = (V, X)$  con longitudes asignadas a sus aristas,  $l : X \rightarrow \mathbb{R}_{\geq 0}$ , el *problema del cartero chino* consiste en encontrar un circuito que pase por cada arista de  $G$  *al menos* una vez de longitud mínima.

- ▶ Sabemos que si el grafo es euleriano, entonces la solución a este problema es un ciclo euleriano.
- ▶ Si el grafo no es euleriano, vamos a tener que recorrer algunas aristas dos veces. Para determinar cuáles, podemos seguir el siguiente procedimiento.



## Problema del cartero chino

1. Definimos  $S = \{v \in V : d(v) \text{ es impar}\}$ , el conjunto de los vértices que tienen grado impar. Llamemos  $s = |S|$  (seguro  $s$  es par).
2. Para cada par de estos vértices,  $v_i, v_j \in S$  calculamos,  $d(v_i, v_j)$ , la longitud del camino mínimo en  $G$ .
3. Armamos el grafo completo pesado,  $K_s$ , con los vértices de  $S$ , donde  $l(v_i, v_j) = d(v_i, v_j)$ .
4. Buscamos  $M$ , un matching perfecto (de cardinal  $s/2$ ) de peso mínimo en este grafo  $K_s$ .
5. Construimos el multigrafo  $G^*$  duplicando en  $G$  las aristas que definen un camino mínimo entre  $v_i$  y  $v_j$  para todas las aristas  $(v_i, v_j) \in M$ .  $G^*$  seguro es euleriano.
6. Buscamos un ciclo euleriano,  $Z$ , en  $G^*$ .
7.  $Z$  es un circuito óptimo para el problema del cartero chino.

## Problema del cartero chino

El único paso que no sabemos hacer es el 4, pero puede realizarse en  $O(s^3)$ . Como el resto de los pasos también son polinomiales, ésto deriva en un algoritmo polinomial para el problema del cartero chino.

Si el grafo es dirigido el procedimiento es similar. En cambio, para grafos mixtos (con aristas y arcos) no se conocen algoritmos polinomiales.