Clase 01 MA3705. 31 de agosto 2020.

Tabla

- Organización e Introducción.
- Optiminación Combinatorial y Algoritmos.
- Grafos: Definiciones básicas.

Equipo Docente

- Cátedra: José Soto (jsoto@dim.uchile.cl)
- Auxiliar: Antonia Labarca
- Auxiliar: Victor Sáez

UCURSOS

Organización e Introducción

Directrices

Directrices/reglas:
Plazo hasta semana 02.

Duración clases

Cátedras: 60 minutos
(+10 si hay presentación)
Auxiliar: 90 minutos.

Ponderación evaluaciones

80 % Tareas.
20 % Trabajo personal.
No hay controles ni examen.

Directrices

Directrices/reglas:
Plazo hasta semana 02.

Duración clases

Cátedras: 60 minutos
(+10 si hay presentación)
Auxiliar: 90 minutos.

Ponderación evaluaciones

80 % Tareas.
20 % Trabajo personal.
No hay controles ni examen.

Tareas

- 4 Tareas en el semestre.
  (2 semanas de plazo c/u)
- Una tarea extra para subir nota.

Directrices

Directrices/reglas:
Plazo hasta semana 02.

Duración clases

Cátedras: 60 minutos
(+10 si hay presentación)
Auxiliar: 90 minutos.

Ponderación evaluaciones

80 % Tareas.
20 % Trabajo personal.
No hay controles ni examen.

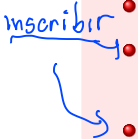
Tareas

- 4 Tareas en el semestre.
  (2 semanas de plazo c/u)
- Una tarea extra para subir nota.

Trabajo personal

6 puntos a llenar de la siguiente manera:

- (3 puntos) Ser escriba una clase
- (3 puntos) Entregar un problema escrito de una lista A de problemas individuales.
- (6 puntos) Presentar en clase solución de una lista B de problemas individuales.



Presentación

Introducción a 3 áreas de la Matemática.

1. Matemáticas Discretas
   Estructuras finitas: Grafos, Redes, Matroides
   ↳ Prerequisito: Algebra Lineal
2. Optimización Combinatorial
   Buscan objetos óptimos en conjuntos finitos.
   ↳ Prerequisito: Optimización (lineal)
3. Algoritmos y Teoría de Computación
   Métodos para resolver problemas.
   ↳ Programación

Eficiencia

Optimización Combinatorial

Optimización Combinatorial

Estudio de Problemas de Optimización en conjuntos finitos

Definición: Problema de Optimización

Un problema $\mathcal{A}$ es un conjunto de instancias con codificación común.

Definición (informal): Instancia

Una instancia es un triple $\mathcal{I} = (\text{Dominio}, f, \text{objetivo})$.

OPT. COMB
DOMINIO : finito.

- Dominio: Conjunto de objetos llamados factibles.
- f: Dominio $\to \mathbb{R}$ es una función a optimizar
- Objetivo: Indica el sentido de optimización (maximizar, minimizar)

Las instancias se codifican de manera implícita o explícita como una palabra (binario).

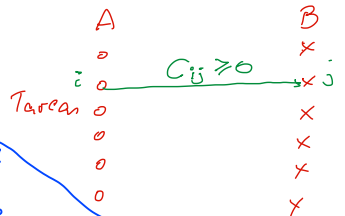
Ejemplo 1: Problema de Asignación

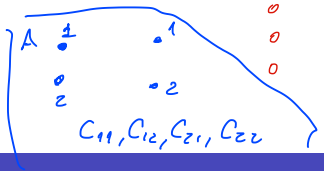
Dados:

- Un conjunto A de n tareas y un conjunto B de n máquinas.
- Costos $c_{ij} \in \mathbb{R}_+$, para cada $i \in A$, $j \in B$: costo de asignar tarea i a máquina j

Tarea:

- Asignar una tarea a cada máquina (de manera biyectiva)
- Minimizando la suma de los costos involucrados.





→ Dominio de una instancia
· funciones biyectivas de A a B.

→ función f. a MINIMIZAR

$$f(x) = \sum_{ij \in X} c_{ij}$$

$$f(\varphi) = \sum_{i \in A} c_{i\varphi(i)}$$

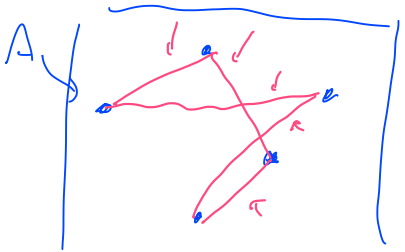
Ejemplo 2: Problema del vendedor viajero en el plano

Dados:

- Un conjunto A de n ciudades (puntos en el plano cartesiano).

Tarea:

- Encontrar un paseo Hamiltonianio que visite a todas las ciudades (es decir, cada ciudad se visita exactamente una vez y se debe terminar en la ciudad que se parte).
- Minimizando el largo total (suma de las distancias recorridas).



ENTRADA:
• Coordenadas (2n números)

DOMINIO: Paseos Hamiltoniano

OBJ: Minimizar largo total

¿Qué es resolver un problema de optimización combinatorial $\mathcal{P}$?

Observación

Si el dominio de una función es finito y no vacío, siempre existe un óptimo de la función.

¿Qué es resolver un problema de optimización combinatorial $\mathcal{P}$?

Observación

Si el dominio de una función es finito y no vacío, siempre existe un óptimo de la función.

Respuesta intuitiva

Método para encontrar solución óptima en cualquier instancia de $\mathcal{P}$ ...

¿Qué es resolver un problema de optimización combinatorial $\mathcal{P}$?

Observación

Si el dominio de una función es finito y no vacío, siempre existe un óptimo de la función.

Respuesta intuitiva

Método para encontrar solución óptima en cualquier instancia de $\mathcal{P}$ ...

o garantizar que no hay solución (declarar DOMINIO vacío)

Método = Algoritmo.

Algoritmo para un Problema P.

Método que recibe una instancia (codificada) como entrada;
ejecuta una secuencia de instrucciones básicas;
devuelve una salida.

Correctitud

Un algoritmo ALG para un problema $\mathcal{P}$ es correcto si al aplicar ALG sobre cualquier instancia de $\mathcal{P}$, el algoritmo:

- termina.
- entrega la solución correcta al terminar.

Ejemplo de un (mal) algoritmo

Fuerza bruta para el problema de asignación

- Generar las $n!$ funciones biyectivas (asignaciones)
- Calcular el costo de cada una
- Devolver la mejor

CORRECTO

- Cada nano segundo generar 1 permutación

Usando este algoritmo en una instancia con $n = 25$

$25!$ ns. $= 491.531.084$ años.

Idea intuitiva de un buen algoritmo

Correcto y Eficiente

Tiempo de resolución depende del tamaño de la entrada.
Debe ser una función de crecimiento pequeño.

Durante el curso formalizaremos el concepto de algoritmo eficiente.
Para llegar a eso necesitamos un lenguaje apropiado para estudiar los problemas de optimización combinatorial.

Grafos: Definiciones básicas

Grafos simples; multigrafos; digrafos

Grafo Simple

$G = (V, E)$. $E \subseteq \binom{V}{2} = \{\{u, v\} : u, v \in V; u \neq v\}\}$

Vértices: $v \in V$. Aristas: $e \in E$.

Multigrafo

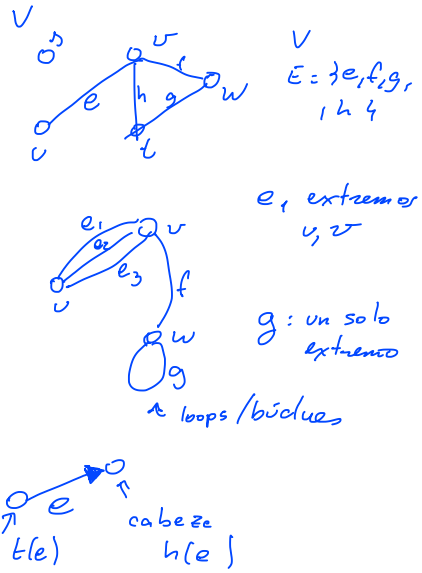
$G = (V, E)$. (vértices y aristas)

Cada arista $e \in E$ posee uno o dos extremos.

Digrafo

$\vec{G} = (V, E)$. (nodos y arcos)

Cada arco $e \in E$ tiene una cola $t(e) \in V$ y una cabeza $h(e) \in V$.



Ejemplos y convenciones adicionales

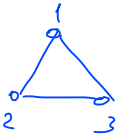
Grafo (Grafos simples)

- $G = (V, E)$. $V(G) \leftarrow$ vertices del grafo
  $E(G) \leftarrow$ aristas del grafo
- Grafos finitos. $|V|, |E| < +\infty$
- Ejemplo de grafo

$$K_n = \left([n], \binom{[n]}{2} = \{\{a,b\} : a,b \in [n], a \neq b\}\right)$$

$$[n] = \{1,2,\dots,n\}$$

K_3



Relaciones y conjuntos en grafos

Sea $G = (V, E)$.

1. extremos.
2. incidente.
3. adyacente.
4. vecinos.

Sean $F \subseteq E$. $U, W \subseteq V$, $U \cap W = \emptyset$, $v \in V$.

1. $F[U, W] :=$
2. $F[U] :=$
3. $\delta_F(W) :=$
4. $N_F(W) :=$
5. $\deg(v)$: (cuidado)