

# QAP



Problema de asignación  
cuadrática

¡HOLA! 

## Somos el Grupo 3

Gregorio Carvajal Expósito

Gema Correa Fernández

Jonathan Fernández Mertanen

Eila Gómez Hidalgo

Elías Méndez García

Alex Enrique Tipán Párraga



# 1.

## ANALISIS DEL PROBLEMA

Explicación



# PROBLEMA

En el QAP, disponemos de  $n$  unidades y  $n$  localizaciones.

En nuestro caso, las localizaciones harán referencia a las habitaciones, y las unidades a los oficinistas.



## MATRIZ DE DISTANCIAS

Habitaciones	H1	H2	H3	H4	H5
H1	0	7	14	20	3
H2	4	0	10	17	49
H3	51	1	0	43	91
H4	7	3	10	0	20
H5	90	101	47	3	0

## MATRIZ DE FLUJOS

Oficinistas	O1	O2	O3	O4	O5
O1	0	4	7	4	1
O2	0	0	10	3	21
O3	0	0	0	47	3
O4	41	21	7	0	9
O5	21	43	32	27	0

# ANÁLISIS DEL PROBLEMA



**Nuestro objetivo es encontrar una asignación de unidades a localizaciones, tal que se minimice el coste dado por la siguiente expresión:**

$$p^* = \min_p \left\{ \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{p(i)p(j)} d_{ij} \right\}$$

- ❑  $f_{p(i)p(j)}$ : es el flujo que circula entre la unidad  $i$  y la  $j$
- ❑  $d_{ij}$ : es la distancia existente entre la localización  $i$  y la  $j$



# 2.

## DISEÑO DE LA SOLUCIÓN

Datos del Problema



## LISTA DE CANDIDATOS

Todas las parejas *unidad<sub>i</sub>*, en *localización<sub>j</sub>*, posibles. En nuestro caso, será una lista formada por parejas de oficinista y habitación.

## CONJUNTO SOLUCIÓN

Cuando todas las unidades hayan sido asignadas a una localización.





## FUNCIÓN SELECCIÓN

Dependerá del beneficio de los candidatos.

$$beneficio_{ij} = |distancia\_potencial_i - flujo\_potencial_j|$$

## FUNCIÓN DE FACTIBILIDAD

Un candidato será factible si aún no se ha asignado esa unidad a una localización, y la localización no se ha asignado a ninguna otra unidad.

# ARCHIVO

## ENTRADA DEL PROGRAMA

5						→ Tamaño del problema
0	7	14	20	3	}	→ Matriz de distancias
4	0	10	17	49		
51	1	0	43	71		
7	3	10	0	20		
90	10	14	7	3		
0	4	7	4	1	}	→ Matriz de flujos
0	0	10	3	21		
0	0	0	47	3		
41	21	7	0	9		
21	43	32	0	27		



# ¿CÓMO OBTENEMOS LA SOLUCIÓN? (I)

Obtenemos los flujos potenciales:



$$\sum_{j=0}^{N-1} f_{ij}$$

O1	O2	O3	O4	O5
fpO1	fpO2	fpO3	fpO4	fpO5

Obtenemos las distancias potenciales:



$$\sum_{j=0}^{N-1} d_{ij}$$

H1	H2	H3	H4	H5
dpH1	dpH2	dpH3	dpH4	dpH5



# ¿CÓMO OBTENEMOS LA SOLUCIÓN? (II)

Obtenemos una matriz de beneficio de cada una de las asignaciones

H1	H2	H3	H4	H5	O1	O2	O3	O4	O5
dpH1	dpH2	dpH3	dpH4	dpH5	fpO1	fpO2	fpO3	fpO4	fpO5

	O1	O2	O3	O4	O5
H1	dpH1-fpO1	dpH1-fpO2	dpH1-fpO3	dpH1-fpO4	dpH1-fpO5
H2	dpH2-fpO1	...	...	...	...
H3	dpH3-fpO1	...	...	...	...
H4	dpH4-fpO1	...	...	...	...
H5	dpH5-fpO1	...	...	...	...



# ¿CÓMO OBTENEMOS LA SOLUCIÓN? (III)

Seleccionamos el mayor de los beneficios y asignamos la pareja a la solución.

	O1	O2	<u>O3</u>	O4	O5
<u>H1</u>	dpH1-fpO1	dpH1-fpO2	dpH1-fpO3	dpH1-fpO4	dpH1-fpO5
H2	dpH2-fpO1	...	...	...	...
H3	dpH3-fpO1	...	...	...	...
H4	dpH4-fpO1	...	...	...	...
H5	dpH5-fpO1	...	...	...	...

i →	H1	H2	H3	H4	H5
v[i] →	O3				



# 3.

## ESQUELETO DEL ALGORITMO

Pseudocódigo



S:Solucion **greedyQAP** (L:localizaciones, U:unidades)

```
para cada localizacion l1 {  
    para cada localizacion l2 {  
        distancia_potencial += distancia_entre(l1, l2)    }  
}
```

```
para cada unidad u1 {  
    para cada unidad u2 {  
        flujo_potencial += flujo_entre(u1, u2)    }  
}
```

LC =  $\emptyset$

```
para cada localizacion l  
    para cada unidad u  
        b = calcular_beneficio(l, u)  
        añadir(LC, l, u, b)
```

Construir  
lista de  
candidatos

```
// Seleccionamos los candidatos para la solución  
mientras (LC  $\neq \emptyset$ ) Y NO solucion(S) hacer  
    x = seleccionar(LC)  
    insertar(S, x)  
    eliminar(LC, x)  
    eliminarNoFactibles(LC, x)
```

**devolver** S

$|distancia\_potencial_i - flujo\_potencial_j|$



# 4.

## FUNCIONAMIENTO DEL ALGORITMO

Veámoslo con un  
ejemplo



# FUNCIONAMIENTO DEL ALGORITMO (I)



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	0	1	2
Localizacion 1	1	0	1
Localizacion 2	1	2	3



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	0	1	2
Localizacion 1	1	0	1
Localizacion 2	1	2	3

	0	1	2
SOLUCION			2



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	0	1	-1
Localizacion 1	1	0	-1
Localizacion 2	-1	-1	-1



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	0	1	-1
Localizacion 1	1	0	-1
Localizacion 2	-1	-1	-1

	0	1	2
SOLUCION		0	2



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	-1	-1	-1
Localizacion 1	1	-1	-1
Localizacion 2	-1	-1	-1



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	-1	-1	-1
Localizacion 1	1	-1	-1
Localizacion 2	-1	-1	-1

	0	1	2
SOLUCION	1	0	2

# FUNCIONAMIENTO DEL ALGORITMO (II)



	Unidad 0	Unidad 1	Unidad 2
Localizacion 0	-1	-1	-1
Localizacion 1	-1	-1	-1
Localizacion 2	-1	-1	-1



**Lista de Candidatos  
Vacía**

**Solución**



0	1	2
1	0	2



# 5.

## APLICACIÓN A UN CASO REAL

Utilidad en el  
mundo real

# Un Problema Real



“

Queremos decidir dónde construir  $n$  fábricas y tenemos  $n$  posibles localizaciones en las que podemos construirlas.

Conocemos las distancias que hay entre cada par de fábricas y también el flujo de materiales que hay de una fábrica a otra.

El problema consiste en decidir dónde construir cada instalación de forma que se minimice el coste de transporte de materiales.





# 6.

## EFICIENCIA TEÓRICA

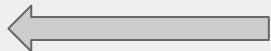
Orden de Eficiencia



```
for i hasta tam_problema  
  for j hasta tam_problema  
    // Calculamos distancias y flujos potenciales [O(1)]
```

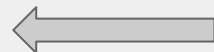
 $O(n^2)$ 

```
for i hasta tam_problema  
  for j hasta tam_problema  
    // Calculamos beneficio de meter oficinista en habitación [O(1)]
```

 $O(n^2)$ 

```
while (NumLC > 0 && tam_solucion < tam_problema) {
```

```
  for i hasta tam_problema  
    for j hasta tam_problema  
      if (LC[i][j] > max)  
        // Buscamos el beneficio más grande y guardamos los índices [O(1)]
```

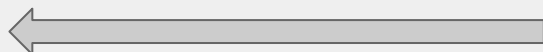
 $O(n^2)$ 

```
  // Añadimos el candidato seleccionado a la solución [O(1)]
```

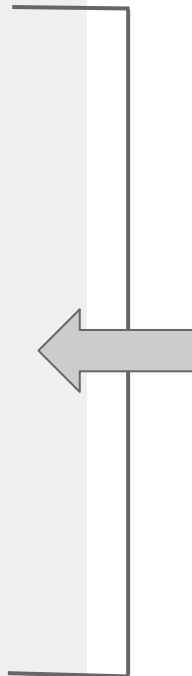
```
  for i hasta tam_problema  
    // Eliminamos los candidatos en la misma fila [O(1)]
```

 $O(n)$ 

```
  for i hasta tam_problema  
    // Eliminamos los candidatos en la misma columna [O(1)]
```

 $O(n)$ 

```
}
```

 $O(n^3)$



**¡GRACIAS A TODOS!**

¿Preguntas?