

# Problemas del Trabajo Práctico

## 03MIAR – Algoritmos de optimización

## Problema 1. Organizar sesiones de doblaje(I)

- Se precisa coordinar el doblaje de una película. Los actores del doblaje deben coincidir en las tomas en las que sus personajes aparecen juntos en las diferentes tomas. Los actores de doblaje cobran toda la misma cantidad por cada día que deben desplazarse hasta el estudio de grabación independientemente del número de tomas que se graben. No es posible grabar más de 6 tomas por día. El objetivo es planificar las sesiones por día de manera que el gasto por los servicios de los actores de doblaje sea el menor posible. Los datos son:

Número de actores: 10

Número de tomas : 30

Actores/Tomas : <https://bit.ly/36D8luK>

- 1 indica que el actor participa en la toma
- 0 en caso contrario

Toma	Actor									
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	0
3	0	1	0	0	1	0	1	0	0	0
4	1	1	0	0	0	0	1	1	0	0
5	0	1	0	1	0	0	0	1	0	0
6	1	1	0	1	1	0	0	0	0	0
7	1	1	0	1	1	0	0	0	0	0
8	1	1	0	0	0	1	0	0	0	0
9	1	1	0	1	0	0	0	0	0	0
10	1	1	0	0	0	1	0	0	1	0
11	1	1	1	0	1	0	0	1	0	0
12	1	1	1	1	0	1	0	0	0	0
13	1	0	0	1	1	0	0	0	0	0
14	1	0	1	0	0	1	0	0	0	0
15	1	1	0	0	0	0	1	0	0	0
16	0	0	0	1	0	0	0	0	0	1
17	1	0	1	0	0	0	0	0	0	0
18	0	0	1	0	0	1	0	0	0	0
19	1	0	1	0	0	0	0	0	0	0
20	1	0	1	1	1	0	0	0	0	0
21	0	0	0	0	0	1	0	1	0	0
22	1	1	1	1	0	0	0	0	0	0
23	1	0	1	0	0	0	0	0	0	0
24	0	0	1	0	0	1	0	0	0	0
25	1	1	0	1	0	0	0	0	0	1
26	1	0	1	0	1	0	0	0	1	0
27	0	0	0	1	1	0	0	0	0	0
28	1	0	0	1	0	0	0	0	0	0
29	1	0	0	0	1	1	0	0	0	0
30	1	0	0	1	0	0	0	0	0	0



## Problema 2. Organizar los horarios de partidos de La Liga(I)

- Desde la La Liga de fútbol profesional se pretende organizar los horarios de los partidos de liga de cada jornada. Se conocen algunos datos que nos deben llevar a diseñar un algoritmo que realice la asignación de los partidos a los horarios de forma que **maximice la audiencia**.
- Los horarios disponibles se conocen a priori y son los siguientes:

Viernes	20
Sábado	12,16,18,20
Domingo	12,16,18,20
Lunes	20

## Problema 2. Organizar los horarios de partidos de La Liga(II)

- En primer lugar, se clasifican los equipos en tres categorías según el número de seguidores( que tiene relación directa con la audiencia). Hay **3 equipos** en la **categoría A**, **11 equipos** de **categoría B** y **6 equipos** de **categoría C**.
- Se conoce estadísticamente la audiencia que genera cada partido según los equipos que se enfrentan y en horario de sábado a las 20h (el mejor en todos los casos)

	Categoría A	Categoría B	Categoría C
Categoría A	2 Millones	1,3 Millones	1 Millones
Categoría B		0.9 Millones	0.75 Millones

## Problema 2. Organizar los horarios de partidos de La Liga(III)

- Si el horario del partido no se realiza a las 20 horas del sábado se sabe que se reduce según los coeficientes de la siguiente tabla
- Debemos asignar obligatoriamente siempre un partido el viernes y un partido el lunes

	<b>Viernes</b>	<b>Sábado</b>	<b>Domingo</b>	<b>Lunes</b>
12h	-	0.55	0.45	-
16h	-	0.7	0.75	-
18h	-	0.8	0.85	-
20h	0.4	1	1	0.4

## Problema 2. Organizar los horarios de partidos de La Liga(IV)

- Es posible la coincidencia de horarios, pero en este caso la audiencia de cada partido se verá afectada y se estima que se reduce en porcentaje según la siguiente tabla dependiendo del número de coincidencias:

Coincidencias	-%
0	0%
1	25%
2	45%
3	60%
4	70%
5	75%
6	78%
7	80%
8	80%

## Problema 2. Organizar los horarios de partidos de La Liga(IV)

Los cálculos asociados a una jornada de ejemplo se realizan según se muestra en la siguiente tabla:

Partido	Categorías	Horario	Base(Mill.)	Ponderación	Base*Ponderación	Corrección Coincidencia
Celta - Real Madrid	B-A	V20	1,3	0,4	0,52	0,52
Valencia - R. Sociedad	B-A	S12	1,3	0,55	0,72	0,72
Mallorca - Eibar	C-C	S16	0,47	0,7	0,33	0,33
Athletic - Barcelona	B-A	S18	1,3	0,8	1,04	1,04
Leganés - Osasuna	C-C	S20	0,47	1	0,47	0,47
Villarreal - Granada	<b>B-C</b>	<b>D16</b>	<b>0,75</b>	<b>0,75</b>	<b>0,56</b>	<b>0,42</b>
Alavés - Levante	<b>B-B</b>	<b>D16</b>	<b>0,9</b>	<b>0,75</b>	<b>0,68</b>	<b>0,51</b>
Espanyol - Sevilla	B-B	D18	0,9	0,85	0,77	0,77
Betis - Valladolid	B-C	D20	0,75	1	0,75	0,75
Atlético - Getafe	B-B	L20	0,9	0,4	0,36	0,36

Total: 5,88

$$=0,56 \times 0,75$$

$$=0,68 \times 0,75$$

## Problema 3. Combinar cifras y operaciones

- El problema consiste en **analizar** el siguiente problema y **diseñar** un algoritmo que lo **resuelva**.
- Disponemos de las 9 cifras del 1 al 9 (excluimos el cero) y de los 4 signos básicos de las operaciones fundamentales: suma(+), resta(-), multiplicación(\*) y división(/)
- Debemos **combinarlos alternativamente sin repetir ninguno de ellos** para obtener una cantidad dada. Un ejemplo sería para obtener el 4:

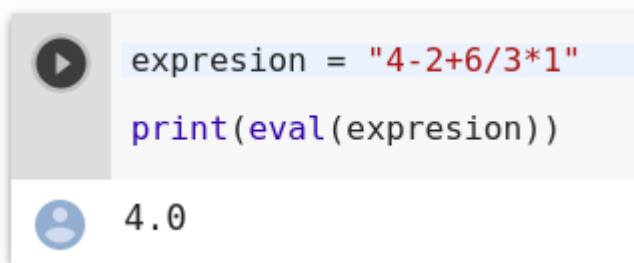
$$4+2-6/3*1 = 4$$





## Problema 3. Combinar cifras y operaciones

- Debe analizarse el problema para encontrar todos los **valores enteros** posibles planteando las siguientes cuestiones:
  - ¿Qué valor **máximo y mínimo** se pueden obtener según las condiciones del problema?
  - ¿Es posible encontrar **todos los valores enteros posibles** entre dicho mínimo y máximo ?
- Nota: Es posible usar la función de python “**eval**” para evaluar una expresión:



```
expression = "4-2+6/3*1"  
print(eval(expression))  
4.0
```



# Problema del Trabajo Práctico

## ACTIVIDAD FORMATIVA

Videoconferencias

Recursos y materiales

Actividades

Mis calificaciones

## COMUNICACIÓN

Anuncios

Foro

Desarrollar algoritmos con la técnica de búsqueda aleatoria  
Desarrollar algoritmos con la técnica de búsqueda local  
Desarrollar algoritmos con la técnica de recocido simulado(simulated annealing)(SA)  
Desarrollar algoritmos con la técnica de colonia de hormigas(ACO)  
Desarrollo de algoritmos genéticos(AG)



### Trabajo Práctico

Desarrollar, modelar y analizar algoritmos según diferentes técnicas para resolver el problema planteado en la asignatura.

Haz una copia de la platilla siguiente de Google Colab para realizar el trabajo:

<https://colab.research.google.com/drive/1NVFHsnmrE-wFLX8y1SC3tKlh2et5FOz8>

Al finalizar, genera el .pdf y adjuntalo como respuesta.

Plantilla: <https://colab.research.google.com/drive/1NVFHsnmrE-wFLX8y1SC3tKlh2et5FOz8>

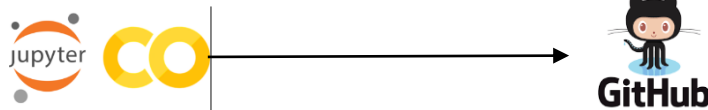
## Problema del Trabajo Práctico

- ¡Importante!. Además de resolver el problema será necesario responder algunas preguntas relacionadas con:
  - Complejidad
  - Justificar la estructura de datos elegida
  - Generar y probar con diferentes juegos de datos de entrada
- Fecha límite de entrega 1ª convocatoria: **23/02/2024**
- Fecha límite de entrega 2ª convocatoria: **22/03/2024**

## Problema del Trabajo Práctico. Entregable

30%

- Generar un Notebook en GitHub (carpeta SEMINARIO)



- Entrega de documento .HTML con en Notebook ()

The screenshot shows a web interface with a sidebar on the left and a main content area. The sidebar has two sections: 'ACTIVIDAD FORMATIVA' with links to 'Videoconferencias', 'Recursos y materiales', 'Actividades', and 'Mis calificaciones'; and 'COMUNICACIÓN' with links to 'Anuncios' and 'Foro'. The main content area has a list of topics: 'Desarrollar algoritmos con la técnica de búsqueda aleatoria', 'Desarrollar algoritmos con la técnica de búsqueda local', 'Desarrollar algoritmos con la técnica de recocido simulado(simulated annealing(SA))', 'Desarrollar algoritmos con la técnica de colonia de hormigas(ACO)', and 'Desarrollo de algoritmos genéticos(AG)'. Below this list, a red box highlights the 'Trabajo Práctico' section, which contains the text: 'Desarrollar, modelar y analizar algoritmos según diferentes técnicas para resolver el problema planteado en la asignatura. Haz una copia de la plantilla siguiente de Google Colab para realizar el trabajo: <https://colab.research.google.com/drive/7NVTfmmrE-wFLX8vISQ3kXh2et5FQe8> Al finalizar, genera el .pdf y adjúntalo como respuesta.' A red arrow points to the URL.

# Problema del Trabajo Práctico. Entregable

- Plantilla para el documento

<https://colab.research.google.com/drive/1NVFHsnmrE-wFLX8y1SC3tKlh2et5FOz8>

30%

Branch: master 03MAIR---Algoritmos-de-Optimizacion---2019 / SEMINARIO / Seminario(plantilla).ipynb Find file Copy path

raul27868 Creado con Colaboratory 3dd96ea 4 minutes ago

1 contributor

417 lines (417 sloc) 8.96 KB

Open in Colab

### Algoritmos de optimización - Seminario

Nombre y Apellidos:  
Url: <https://github.com/.../03MAIR---Algoritmos-de-Optimizacion---2019/tree/master/SEMINARIO>  
Problema:

1. Elección de grupos de población homogéneos
2. Organizar los horarios de partidos de La Liga
3. Combinar cifras y operaciones

Descripción del problema:(copiar enunciado)

....

(\*) La respuesta es obligatoria

In [0]:

(\*)¿Cuantas posibilidades hay sin tener en cuenta las restricciones?

¿Cuantas posibilidades hay teniendo en cuenta todas las restricciones.

# Problema del Trabajo Práctico. Entregable



30%

- Cabecera

## Algoritmos de optimización - Seminario

Nombre y Apellidos:

Url: <https://github.com/.../03MAIR---Algoritmos-de-Optimizacion---2019/tree/master/SEMINARIO>

Problema:

- ~~1. Elección de grupos de población homogéneos~~
- ~~2. Organizar los horarios de partidos de La Liga~~
3. Combinar cifras y operaciones

Descripción del problema: (copiar enunciado)

...

Añadir texto del enunciado

(\*) La respuesta es obligatoria

[ ]

## Problema del Trabajo Práctico. Entregable

- Pregunta – Respuesta (texto + Python)

30%

(\*)¿Cuantas posibilidades hay sin tener en cuenta las restricciones?  
¿Cuantas posibilidades hay teniendo en cuenta todas las restricciones.

Respuesta

	Texto
[ ]	Código python

Obligatoria

# Problema del Trabajo Práctico. Entregable. Ejemplo



## Pregunta – Respuesta (texto + Python)

(\*)¿Cuántas posibilidades hay sin tener en cuenta las restricciones?

¿Cuántas posibilidades hay teniendo en cuenta todas las restricciones.

Respuesta:

Para calcular el número posible de soluciones es necesario algo de combinatoria y saber contar.

Suponemos que el número de ciudades es  $N$  y que partimos de una ciudad dada. Podemos suponer un procedimiento que vaya construyendo todas las soluciones.

Para el primer viaje disponemos de  $N-1$  ciudades ya que debemos eliminar la ciudad de partida como posible candidata. Para la segunda ciudad a visitar disponemos de  $N-2$  posibilidades. Por tanto ya tenemos  $(N-1) \times (N-2)$  para visitar 2 ciudades.

Si seguimos el razonamiento deducimos que hay  $(N-1)!$  (factorial de  $N-1$ ) posibilidades.

Puesto que el camino es circular (comienza y termina en la misma ciudad) debemos tener en cuenta que cada ruta tiene una ruta inversa semejante. La primera se convierte en la última, la segunda en la penúltima y así sucesivamente. Por tanto sin no queremos tener en cuenta esta repetición en total tenemos  $(N-1)!/2$



## Problema del Trabajo Práctico. Preguntas(1/3)



### Pregunta – Respuesta (texto + Python)

- (\*)¿Cuántas posibilidades hay sin tener en cuenta las restricciones?
- ¿Cuántas posibilidades hay teniendo en cuenta todas las restricciones?
- (\*) ¿Cuál es la estructura de datos que mejor se adapta al problema? Argumenta la respuesta  
(Es posible que hayas elegido una al principio y veas la necesidad de cambiar, argumenta)
- (\*)¿Cuál es la función objetivo?
- (\*)¿Es un problema de maximización o minimización?

## Problema del Trabajo Práctico. Preguntas(2/3)



### Pregunta – Respuesta (texto + Python)

- Diseña un algoritmo para resolver el problema por fuerza bruta
- Calcula la complejidad del algoritmo por fuerza bruta
- (\*)Diseña un algoritmo que mejore la complejidad del algoritmo por fuerza bruta. Argumenta porque crees que mejora el algoritmo por fuerza bruta
- (\*)Calcula la complejidad del algoritmo
- Según el problema (y tenga sentido), diseña un juego de datos de entrada aleatorio.

## Problema del Trabajo Práctico. Preguntas(3/3)



### Pregunta – Respuesta (texto + Python)

- Aplica el algoritmo al juego de datos aleatorio generado.
- Enumera las referencias que has utilizado(si ha sido necesario) para llevar a cabo el trabajo
- Describe brevemente en unas líneas como crees que es posible avanzar en el estudio del problema. Ten en cuenta incluso posibles variaciones del problema y/o variaciones al alza del tamaño.

## Problema del Trabajo Práctico. Evaluación.

Total 13 cuestiones:

6 obligatorias(\*) , aseguran 7/10

- 7 opcionales , añaden 2 puntos más: 9/10
  - 1 punto por presentación, descripción, ...
    - lenguaje claro
    - código comentado
    - acompaña ilustraciones(imágenes) si es necesario
- ...

Fecha límite de entrega 1ª convocatoria: 23/02/2024

Fecha límite de entrega 2ª convocatoria: 22/03/2024



# Ampliación de conocimientos y habilidades

## ■ Bibliografía

-Brassard, G., y Bratley, P. (1997). Fundamentos de algoritmia. *ISBN 13: 9788489660007*

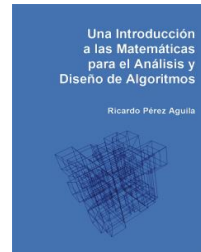
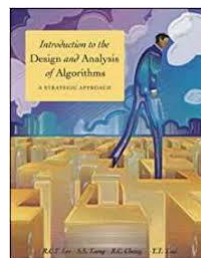
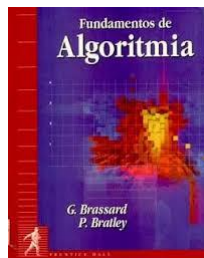
-Guerequeta, R., y Vallecillo, A. (2000). Técnicas de diseño de algoritmos.

<http://www.lcc.uma.es/~av/Libro/indice.html>

-Lee, R. C. T., Tseng, S. S., Chang, R. C., y Tsai, Y. T. (2005). Introducción al diseño y análisis de algoritmos. *ISBN 13: 9789701061244*

- Pérez Aguila, R. (2012). Una introducción a las matemáticas para el análisis y diseño de algoritmos.

*ISBN 13: 9781413576474.* <https://tinyurl.com/yzlt5oed>

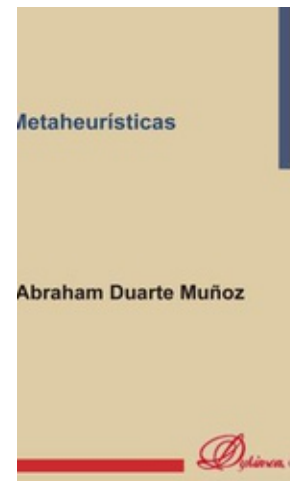
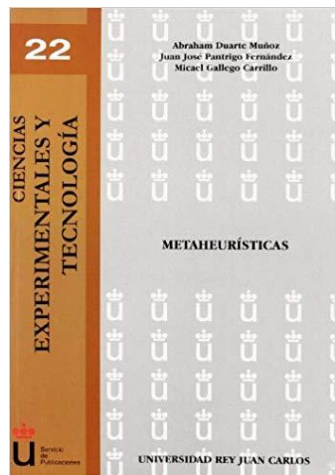


# Ampliación de conocimientos y habilidades

## ■ Bibliografía

- Duarte, A. (2008). Metaheurísticas. Madrid: Dykinson.

<https://elibro-net.universidadviu.idm.oclc.org/es/ereader/universidadviu/35696>



## ¿Preguntas?



# Gracias

[juanfrancisco.vallalta@professor.universidadviu.com](mailto:juanfrancisco.vallalta@professor.universidadviu.com)