

# Algorítmica - Practica 4: Backtracking

*A. Herrera, A. Moya, I. Sevillano, J.L. Suarez*

*16 de mayo de 2015*

## Contents

<b>1</b>	<b>Organización de la práctica</b>	<b>2</b>
<b>2</b>	<b>Problema 5: Estación de ITV</b>	<b>3</b>
2.1	Enunciado del problema . . . . .	3
2.2	Solución teórica . . . . .	3
<b>3</b>	<b>Referencias</b>	<b>3</b>

# 1 Organización de la práctica

La práctica 4 consiste en el desarrollo de algoritmos basados en backtracking que consigan la solución óptima de los problemas propuestos. Usualmente las soluciones obtenidas serán exponenciales ya que intentaremos resolver problemas NP. La poda nos permitirá conseguir en la práctica resultados aceptables.

Nuestro grupo debe resolver el problema 5.

Para cada problema resuelto se sigue la siguiente estructura:

- Enunciado del problema
- Resolución teórica del problema (con una subsección por algoritmo)
- Análisis empírico. Análisis de la eficiencia híbrida

En este último apartado se proporcionan gráficas con los resultados de los algoritmos y un análisis de la eficiencia híbrida para los mismos.

Los algoritmos se han ejecutado sobre un ordenador con las siguientes características:

- **Marca:** Toshiba
- **RAM:** 8 GB
- **Procesador:** Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz

El código, los resultados de las ejecuciones, las gráficas y los pdf asociados se pueden encontrar en [GitHub](#).

## 2 Problema 5: Estación de ITV

### 2.1 Enunciado del problema

Una estación de ITV consta de  $m$  líneas de inspección de vehículos iguales. Hay un total de  $n$  vehículos que necesitan inspección. En función de sus características, cada vehículo tardará en ser inspeccionado un tiempo  $t_i$ ,  $i = 1, \dots, n$ . Se desea encontrar la manera de atender a los  $n$  vehículos y acabar en el menor tiempo posible. Diseñar e implementar un algoritmo vuelta atrás que determine cómo asignar los vehículos a las líneas. Mejorarlo usando alguna técnica de poda. Realizar un estudio empírico de la eficiencia de los algoritmos.

### 2.2 Solución teórica

La formulación del problema recuerda al problema 5 de la práctica de algoritmos voraces. Tenemos varias máquinas que realizan trabajos que requieren un tiempo predeterminado. Este tipo de problemas suelen clasificarse como **scheduling problems**. Sin embargo, la función objetivo a optimizar es diferente. En el problema de la práctica anterior se pedía optimizar el tiempo medio de espera de los clientes mientras que en este caso debemos realizar todos los trabajos en el menor tiempo posible.

Es un problema NP Hard [^:wikipedia-scheduling]. Por tanto, queremos conseguir la mejor solución exponencial posible. Antes de nada hacemos la siguiente observación, dada una máquina, no nos importa el orden en el que realice sus trabajos asignados ya que en cualquier caso el tiempo en el que está trabajando es el mismo. Esto da lugar a la siguiente proposición:

#### Proposición 1.

El número de formas en las que se pueden asignar los trabajos a las máquinas es:

$$CR_n^m (n + m - 1n)$$

**Demostración.** Cada posible asignación podemos verla como que a cada trabajo se le asigna una máquina, pudiendo asignar una máquina a varios trabajos. Esto no es otra cosa que las combinaciones con repetición de  $m$  elementos tomados de  $n$  en  $n$ .

■

Otra cosa que debemos observar es que no nos importa cuál sea la máquina que realiza un determinado conjunto de trabajo. Por ejemplo, si tuviésemos dos máquinas,  $A$  y  $B$ , y dos trabajos, 1 y 2, da igual que la máquina  $A$  haga el trabajo 1 y la máquina  $B$  haga el trabajo 2 a que  $A$  realice el 2 y  $B$  el 1. Las máquinas son igual de eficientes por lo que el tiempo en el que se habrán terminado todos los trabajos será el mismo en ambos casos.

El razonamiento previo nos reduce a estudiar ..... posibles asignaciones de trabajos. Otro detalle importante es que

## 3 Referencias

[^:wikipedia-scheduling]() Scheduling Problem