



UNIVERSIDAD  
DE GRANADA

**E.T.S. DE INGENIERÍAS INFORMÁTICA y DE  
TELECOMUNICACIÓN**

**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

**Algorítmica**

**Guión de Prácticas**

**Práctica 3: Algoritmos Greedy**

**Grado en Ingeniería Informática**

# 1. Objetivo

El objetivo de esta práctica es que el estudiante aprecie la utilidad de los métodos voraces (greedy) para resolver problemas de forma muy eficiente, en algunos casos obteniendo soluciones óptimas y en otros aproximaciones. Para ello deberá resolver cada problema solo o en equipo. En ese último caso, cada equipo debería exponer y defender su propuesta en el seno de dicho equipo, con el fin de ir aprendiendo a expresarse en público. Adicionalmente, en todos los casos, ya sea en solitario o en equipo, se propone implementar algoritmos greedy para resolver el Problema del Viajante de Comercio, a fin de comprobar la eficacia de las heurísticas greedy.

## 2. Problemas

### 2.1. Contenedores en un barco

Se tiene un buque mercante cuya capacidad de carga es de  $K$  toneladas y un conjunto de contenedores  $c_1, \dots, c_n$  cuyos pesos respectivos son  $p_1, \dots, p_n$  (expresados también en toneladas). Teniendo en cuenta que la capacidad del buque es menor que la suma total de los pesos de los contenedores:

- Diseñe un algoritmo que maximice el número de contenedores cargados, y demuestre su optimalidad.
- Diseñe un algoritmo que intente maximizar el número de toneladas cargadas.

### 2.2. Minimizando el número de visitas al proveedor

Un granjero necesita disponer siempre de un determinado fertilizante. La cantidad máxima que puede almacenar la consume en  $r$  días, y antes de que eso ocurra necesita acudir a una tienda del pueblo para abastecerse. El problema es que dicha tienda tiene un horario de apertura muy irregular (solo abre determinados días). El granjero conoce los días en que abre la tienda, y desea minimizar el número de desplazamientos al pueblo para abastecerse.

- Diseñar un algoritmo greedy que determine en qué días debe acudir al pueblo a comprar fertilizante durante un periodo de tiempo determinado (por ejemplo durante el siguiente mes).
- Demostrar que el algoritmo encuentra siempre la solución óptima.

### 2.3. Minimizando el tiempo medio de acceso

Sean  $n$  programas  $P_1, P_2, \dots, P_n$  que hay que almacenar en una cinta. El programa  $P_i$  requiere  $s_i$  kilobytes de espacio y la cinta es suficientemente larga para almacenar todos los programas.

Se sabe con qué frecuencia se utiliza cada programa: una fracción  $\pi_i$  de las solicitudes afecta al programa  $P_i$  (y por tanto  $\sum_{i=1}^n \pi_i = 1$ ). Los datos se almacenan en la cinta con densidad

constante y la velocidad de la cinta también es constante. Una vez que se carga el programa, la cinta se rebobina hasta el principio.

Si los programas se almacenan por orden  $i_1, i_2, \dots, i_n$  el tiempo medio requerido para cargar un programa es, por tanto:

$$\hat{T} = c \sum_{j=1}^n \left[ \pi_{i_j} \sum_{k=1}^j s_{i_k} \right]$$

donde la constante  $c$  depende de la densidad de grabación y de la velocidad de la cinta.

Se desea minimizar  $\hat{T}$  empleando un algoritmo voraz. Demuestre la optimalidad del algoritmo o encuentre un contraejemplo que muestre que el algoritmo no es óptimo para los siguientes criterios de selección:

- Programas en orden no decreciente de  $s_i$ .
- Programas en orden no creciente de  $\pi_i$ .
- Programas en orden no creciente de  $\pi_i/s_i$ .

## 2.4. Reparaciones

Un electricista necesita hacer  $n$  reparaciones urgentes, y sabe de antemano el tiempo que le va a llevar cada una de ellas: en la tarea  $i$ -ésima tardará  $t_i$  minutos.

Como en su empresa le pagan dependiendo de la satisfacción del cliente y esta es inversamente proporcional al tiempo que tardan en atenderles, necesita decidir el orden en el que atenderá los avisos para minimizar el tiempo medio de atención de los clientes (desde el inicio hasta que su reparación es efectuada).

Diseñar un algoritmo greedy para resolver esta tarea. Demostrar que el algoritmo obtiene la solución óptima.

- Modificar el algoritmo anterior para el caso de una empresa en la que se disponga de los servicios de más de un electricista.