



**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 4: Algoritmos de Vuelta Atrás (Backtracking) y  
de Ramificación y Poda (Branch and Bound)**

Curso 2020-2021

Grado en Informática

# Objetivo

El objetivo de esta práctica es que el estudiante aprecie la potencia de los métodos vuelta atrás y de ramificación y poda para resolver problemas, pero también comprenda sus limitaciones. Para ello cada equipo de estudiantes deberá diseñar e implementar un algoritmo de vuelta atrás para resolver uno de los problemas (asignado al azar) que se describen a continuación en las secciones 1–5.

Adicionalmente, todos los equipos deberán diseñar e implementar un algoritmo de ramificación y poda para el problema descrito en la sección 6.

## 1. Elección de árbitros

En una liga de cierto deporte participan  $n$  equipos (suponemos que  $n$  es par). En cada jornada se juegan  $n/2$  partidos, que enfrentan a dos equipos, dirigidos por un árbitro. Existen  $m$  árbitros disponibles, siendo  $m > n/2$ . Cada equipo  $i$  valora a cada árbitro  $j$  con una puntuación  $p(i, j)$  entre 1 y 10, indicando su preferencia por ese árbitro (un valor alto indica que le gusta el árbitro y un valor bajo que no le gusta). El objetivo es, dada una jornada concreta con unos emparejamientos entre equipos, asignar un árbitro distinto a cada partido, de manera que se maximice la puntuación total de los árbitros asignados, teniendo en cuenta las preferencias de todos los equipos. Esto se medirá mediante la suma de los productos de las puntuaciones asignadas por cada equipo al árbitro asignado a cada partido.

Diseñad e implementad un algoritmo de vuelta atrás para resolver el problema. Mejoradlo usando alguna cota que mejore la poda. Realizar un estudio empírico de la eficiencia de los algoritmos (incluyendo también fuerza bruta).

## 2. Solitario

Se trata de un juego (solitario) donde se colocan 32 piezas iguales en un tablero de 33 casillas, tal y como se indica en la figura siguiente (las “x” corresponden a posiciones no válidas):

x	x	o	o	o	x	x
x	x	o	o	o	x	x
o	o	o	o	o	o	o
o	o	o		o	o	o
o	o	o	o	o	o	o
x	x	o	o	o	x	x
x	x	o	o	o	x	x

Solo se permiten movimientos de las piezas en vertical y horizontal. Una pieza solo puede moverse saltando sobre otra y situándose en la siguiente casilla, que debe estar vacía. La pieza sobre la que se salta se retira del tablero. Se consigue terminar con éxito el juego cuando queda una sola pieza en la posición central del tablero (la que estaba inicialmente vacía).

Diseñad e implementad un algoritmo vuelta atrás que encuentre una serie de movimientos para llegar con éxito al final del juego.

### 3. Diviando en dos grupos equilibrados

Se desea dividir un conjunto de  $n$  personas para formar dos equipos que competirán entre sí. Cada persona tiene un cierto nivel de competición, que viene representado por una puntuación (un valor numérico entero). Con el objeto de que los dos equipos tengan una capacidad de competición similar, se pretende construir los equipos de forma que la suma de las puntuaciones de sus miembros sea la misma. Diseñad e implementad un algoritmo vuelta atrás para resolver, si es posible, este problema. Modificadlo para que, incluso cuando no haya solución, encuentre la división en dos equipos que minimice la diferencia entre la suma de las puntuaciones de los dos. Se puede suponer que la suma de las puntuaciones de todas las personas es un número par.

Realizar también un estudio empírico de la eficiencia de los algoritmos.

### 4. Minimizar la suma de cuadrados sumando $n$

Dado un entero positivo  $n$ , se desea encontrar un conjunto de números enteros positivos distintos  $\{x_1, x_2, \dots, x_m\}$  tales que su suma sea  $n$  ( $\sum_{i=1}^m x_i = n$ ) y la suma de sus cuadrados sea mínima (se pretende minimizar  $\sum_{i=1}^m x_i^2$ ). Nota:  $m$  no es un parámetro de entrada.

Diseñad e implementad un algoritmo de vuelta atrás para resolver este problema. Mejoradlo usando alguna cota que mejore la poda. Realizar también un estudio empírico de la eficiencia.

### 5. Saltando por una matriz

Dada una matriz  $m$  de tamaño  $n \times n$ , que contiene números enteros entre 1 y  $n - 1$ , se quiere encontrar un camino desde una posición origen (p.e. la posición  $(0, 0)$ ) hasta una posición destino  $(a, b)$  saltando a determinadas posiciones. Las reglas de movimiento son: (1) hay que moverse en horizontal o en vertical (moverse por la misma fila o la misma columna, no se puede mover en diagonal; si estamos en la posición  $(i, j)$ , entonces la posición a la que saltamos tiene que estar en la fila  $i$  o en la columna  $j$ ); (2) el número exacto de casillas que saltamos estando en la posición  $(i, j)$  viene determinado por el valor  $m(i, j)$  (de modo que los 4 movimientos en principio posibles son a las posiciones  $(i + m(i, j), j)$ ,  $(i - m(i, j), j)$ ,  $(i, j + m(i, j))$  y  $(i, j - m(i, j))$ ). Por ejemplo si  $n = 10$ , estamos en la posición  $(1, 6)$  y  $m(1, 6) = 6$ , entonces podemos saltar a las posiciones  $(7, 6)$  y  $(1, 0)$ , los otros dos movimientos en principio posibles no serían válidos en este caso.

Diseñad e implementad un algoritmo de backtracking para resolver el problema. Modificadlo para que encuentre el camino con menor número de saltos.

### 6. Ramificación y poda: Recipientes

El problema de los recipientes, ya expuesto en la práctica anterior sobre algoritmos voraces, consiste en:

## 6.1. Recipientes

Tenemos  $n$  objetos de pesos  $w_1, w_2, \dots, w_n$ , y un número ilimitado de recipientes iguales con capacidad máxima  $R$  (siendo  $w_i \leq R, \forall i$ ). Los objetos se deben meter en los recipientes sin partirlos, y sin superar su capacidad máxima. Se busca el mínimo número de recipientes necesarios para colocar todos los objetos, y conocer cuál es la asignación de objetos a los recipientes para esa configuración óptima.

## 6.2. Tareas

En esta práctica se va a abordar este problema mediante un algoritmo de ramificación y poda que garantice encontrar la solución óptima, no una solución aproximada como en el caso de los algoritmos voraces.

- Diseñad e implementad un algoritmo que utilice la técnica de ramificación y poda para resolver el problema de los recipientes. Se debe definir una cota local optimista (inferior en este caso) del valor que podría obtenerse al explorar el subárbol con raíz en un nodo dado, y utilizarla, además de para podar (comparándola con la cota global, superior en este caso), para guiar el proceso de búsqueda (usando la estrategia LC).
- Además de devolver el costo de la solución encontrada (y la asignación correspondiente), se deben de obtener también resultados relativos a complejidad: número de nodos expandidos, tamaño máximo de la cola con prioridad de nodos vivos, número de veces que se realiza la poda y el tiempo empleado en resolver el problema.
- *Opcionalmente* utilizad también un algoritmo de vuelta atrás para resolver este mismo problema (utilizando también la cota definida antes).
- Utilizad el algoritmo de exploración exhaustiva de soluciones de la práctica anterior para comprobar, con tamaños de problema pequeños, que las soluciones encontradas por los algoritmos de exploración de grafos son realmente óptimas.

## Nota importante

Se debe de entregar una memoria detallada con todas las tareas realizadas (de las dos partes de la práctica). En todos los casos se debe especificar claramente, además de los algoritmos, la representación del problema, la representación de la solución, las restricciones explícitas e implícitas, el árbol de exploración, así como las posibles cotas a utilizar. Asimismo se darán detalles del estudio empírico realizado en cada parte. Se debe incluir el código de todos los programas desarrollados. El informe debe entregarse en formato pdf. En el informe deben aparecer los nombres de todos los miembros del equipo que hayan participado en la realización de la práctica, así como el porcentaje de participación de cada miembro del equipo.