



**UNIVERSIDAD
DE GRANADA**

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

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

Doble Grado en Ingeniería Informática y Matemáticas

Curso 2022-2023

Algorítmica

Guión de Prácticas

Práctica 4: Exploración en grafos

Índice

1. Enunciado de los problemas	1
1.1. Problema del viajante de comercio	1
1.2. Problema de la mochila	2
1.3. Problema de asignación de tareas	2
1.4. Problema del electricista con penalizaciones	3
1.5. Problema de la cena de gala	3
2. Trabajo a realizar	3
2.1. Tareas	3
2.2. Memoria	5
3. Evaluación de la práctica	6

Objetivo

El objetivo de esta práctica es que el estudiante comprenda y asimile el funcionamiento de las técnicas de resolución de problemas basadas en exploración de grafos: *Backtracking* y *Branch and Bound*. Esta práctica será realizada por grupos de alumnos. A cada grupo se le asignará un problema para el que tendrá que diseñar varios algoritmos según las indicaciones incluidas en este guión.

En la sección que sigue se muestran los enunciados de los problemas a considerar.

1. Enunciado de los problemas

En esta sección se incluyen los enunciados de los problemas que tienen que resolver cada grupo de alumnos.

1.1. Problema del viajante de comercio

Un agente commercial tiene que visitar n ciudades. Se proporciona una matriz cuadrada D , de dimensiones $n \times n$ tal que $d_{i,j}$ indica el coste de viajar desde la ciudad i a la j . El agente debe realizar un recorrido visitando todas

las ciudades y finalizando en la misma ciudad desde la que partió. Además sólo puede visitar cada ciudad una sola vez. El coste global de un recorrido dado es la suma de los costes de todos los traslados.

El objetivo es encontrar un recorrido que cumpla las restricciones establecidas y tenga el menor coste posible.

1.2. Problema de la mochila

Se dispone de n objetos cada uno está caracterizado por dos propiedades, un peso, p_i y un beneficio b_i , con $p_i, b_i \in \mathbb{R}^+, \forall i \in \{1, \dots, n\}$. Además se dispone de una mochila que soporta un peso máximo de M .

El objetivo es encontrar una selección de objetos, representada como un vector binario $\mathbf{x} = (x_1, \dots, x_n)$ tal que se obtenga el máximo beneficio, sin exceder el peso máximo que puede soportar la mochila:

$$\begin{aligned} & \text{máx} \sum_{i=1}^n x_i b_i \\ & \text{s.a.} \\ & x_i \in \{0, 1\} \\ & \sum_{i=1}^n x_i p_i \leq M \end{aligned}$$

Formalmente, este enunciado corresponde a la versión básica del problema de la mochila, habitualmente nombrado “Mochila (0,1)”.

1.3. Problema de asignación de tareas

Se dispone de n personas, que han de realizar n tareas. Cada persona debe hacer exactamente una única tarea, y cada tarea debe ser realizada solo por una persona. Se dispone de una matriz cuadrada M , de tamaño $n \times n$. La celda $m_{i,j}$ de la matriz indica el coste de que la persona i -ésima realice la tarea j , con $m_{i,j} \in \mathbb{R}^+$.

El objetivo es encontrar una asignación de tareas a personas, σ , de forma que se minimice el coste de la asignación, definido como:

$$\sum_{i=1}^n m_{i, \sigma(i)},$$

donde $\sigma(i)$ es la tarea asignada a la persona i -ésima.

1.4. Problema del electricista con penalizaciones

Supongamos que un electricista tiene n avisos pendientes, y que cada uno de ellos, i , tiene asociados tres valores (d_i, f_i, p_i) : d_i duración (expresada en días), f_i plazo límite (expresado en días) y p_i penalización (lo que deja de ganar) en caso de que no ejecute dentro del plazo límite establecido (expresada en unidades monetarias).

Por ejemplo, para el caso de cuatro avisos ($n = 4$) podríamos tener los siguientes datos:

Aviso	1	2	3	4
Duración (d)	2	1	2	3
Plazo límite (f)	3	4	4	3
Penalización (p)	50	150	13	10

El objetivo es determinar la fecha de comienzo de cada tarea de forma que la penalización total sea mínima.

1.5. Problema de la cena de gala

Se va a celebrar una cena de gala a la que asistirán n invitados. Todos se van a sentar alrededor de una gran mesa rectangular, de forma que cada individuo tendrá sentados junto a él a dos comensales (uno a su izquierda y otro a su derecha). Las normas de protocolo establecen la *conveniencia* de que dos individuos, en base al cargo que ocupan, se sienten en lugares contiguos. Por simplicidad asumiremos que ese valor de conveniencia se expresa como un entero positivo entre 0 y 1000. El nivel de conveniencia global se define como la suma de todos los niveles de conveniencia de cada invitado con cada uno de los dos comensales sentados a sus dos lados.

El objetivo es sentar a los invitados de forma que el nivel de conveniencia global sea lo mayor posible.

2. Trabajo a realizar

Para alcanzar el objetivo previsto con esta práctica, los alumnos deberán realizar las tareas que se detallan en la sección siguiente, documentando el trabajo realizado en una memoria con el contenido y estructura que se indica en la subsección 2.2.

2.1. Tareas

1. **Leer por completo** este documento y asegurarse de que se entienden perfectamente todas las indicaciones incluidas y de que la memoria que

se entrega es conforme a todas ellas.

2. Estudiar en profundidad el problema asignado, asegurándose de comprender bien las entradas y salidas involucradas, así como las relaciones entre éstas. Deberán identificar el tipo de información que define cada caso del problema y las magnitudes que definen su tamaño.
3. Diseñar e implementar un generador de casos para el problema.

Técnica *Backtracking*

4. Diseñar un algoritmo basado en *Backtracking* para resolver el problema. Diseñar varias funciones de cota.
5. Demostrar o justificar la validez del algoritmo, es decir, que la solución que proporciona es la correcta (óptima, en el caso de problemas de optimización).
6. Implementar el algoritmo y las distintas funciones de cota. Implementarlo de modo que se pueda elegir qué cota se aplica en cada ejecución concreta.
7. Realizar el análisis de la eficiencia teórica, empírica e híbrida los algoritmos.

Técnica *Branch and Bound*

8. Diseñar un algoritmo basado en *Branch and Bound* para resolver el problema. Diseñar varias funciones de cota.
9. Demostrar o justificar la validez del algoritmo, es decir, que la solución que proporciona es la correcta (óptima, en el caso de problemas de optimización).
10. Implementar el algoritmo y las distintas funciones de cota. Implementarlo de modo que se pueda elegir qué cota se aplica en cada ejecución concreta.
11. Realizar el análisis de la eficiencia teórica, empírica e híbrida los algoritmos.
12. Elaborar una memoria que documente el trabajo realizado de acuerdo a las indicaciones incluidas en la subsección 2.2.

2.2. Memoria

Todo el trabajo realizado debe redactarse en una memoria. La estructura de este documento será la siguiente:

1. Portada. Incluyendo las denominaciones de titulación, asignatura y práctica. También el nombre completo de los alumnos que forman el grupo y su dirección de correo electrónico.
2. Autores. Indicar el % del trabajo realizado por cada alumno, especificando qué tareas ha realizado cada uno.
3. Objetivos. Descripción del objetivo de la práctica.
4. Definición del problema. Descripción de los casos usados en la evaluación de la eficiencia. Descripción completa del entorno de análisis: hardware, sistema operativo, compilador, etc. empleados. Descripción del método de medición de tiempos.

Técnica *Backtracking*

5. Algoritmo diseñado. Descripción del algoritmo, con detalle específico de la formulación del problema, representación de las soluciones, el árbol de exploración, restricciones explícitas e implícitas y la función de cota. Demostración o justificación de su validez. Proponer funciones de cota alternativas.
6. Análisis de la eficiencia: teórica, empírica e híbrida.
7. Análisis comparativo del rendimiento y tiempos de las distintas funciones de cota consideradas.

Técnica *Branch and Bound*

8. Algoritmo diseñado. Descripción del algoritmo, con detalle específico de la formulación del problema, representación de las soluciones, el árbol de exploración, restricciones explícitas e implícitas y la función de cota. Demostración o justificación de su validez. Proponer funciones de cota alternativas.
9. Análisis de la eficiencia: teórica, empírica e híbrida.
10. Análisis comparativo del rendimiento y tiempos de las distintas funciones de cota consideradas.

11. Análisis comparativo de las dos técnicas heurísticas en términos de la calidad de las soluciones que construyen y tiempos de ejecución.
12. Conclusiones.

3. Evaluación de la práctica

Esta práctica se realizará por grupos formados por tres alumnos, ya conformados para las prácticas anteriores.

El profesor asignará a cada grupo el problema que deben resolver.

Los alumnos habrán de entregar dos archivos en la actividad correspondiente incluida en la página correspondiente a la asignatura en la plataforma Prado. El primero archivo será la memoria, en formato pdf.

El segundo será el código fuente de las implementaciones realizadas empaquetado en un fichero .zip. El código estará organizado en seis carpetas de nombres:

1. **Generador**: generador de casos del problema asignado.
2. **bt**: algoritmo basado en *Backtracking* resuelve el problema asignado. Incluirá todas las funciones de cota diseñadas.
3. **bb**: algoritmo basado en *Branch and Bound* resuelve el problema asignado. Incluirá todas las funciones de cota diseñadas.

Cada una de estas carpetas incluirá todos los módulos de código fuente necesarios para generar el programa binario correspondiente. Además, en la carpeta raíz habrá un único fichero **Makefile** cuyo objetivo por defecto construirá los tres binarios de nombres **generador**, **bt** y **bb**.

La **fecha límite** para entregar la memoria es el día **15 de mayo de 2023** a las 23:59 horas.

Además de hacer el trabajo y entregar la memoria, cada equipo tendrá que elaborar una breve presentación del trabajo realizado que expondrá públicamente en clase de prácticas, de acuerdo a la convocatoria establecida por el profesor. Es obligatoria la participación de todos los miembros del grupo en la exposición.