



**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 exploración de grafos

Curso 2023-2024

Doble Grado en Informática y Matemáticas

Índice

1. Enunciado de los problemas	2
2. Trabajo a realizar	2
2.1. Tareas	2
3. Memoria	3
4. Evaluación de la práctica	4

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

1. Enunciado de los problemas

P1: Problema del viajante de comercio

Tenemos un conjunto de n ciudades (puntos en un plano), cada una definida por las coordenadas en el mapa (x_i, y_i) , con $i = 1, \dots, n$. La distancia entre dos ciudades viene dada por la distancia euclídea entre sus coordenadas.

$$\text{dist}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

El problema de viajante de comercio consiste en encontrar el orden en el que un viajante, partiendo de la ciudad de origen (por ejemplo (x_1, y_1)) pase por todas y cada una de las ciudades una única vez, para volver a la ciudad de partida, formando un ciclo.

El costo del ciclo será la suma de las distancias que hay entre las ciudades.

El problema original del viajante de comercio consiste en encontrar el ciclo de costo mínimo entre todas las posibilidades existentes.

Aunque este problema es NP-Difícil y por tanto podemos esperar encontrar una solución óptima al mismo, aunque el tiempo no será viable para tamaño de problemas grandes.

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 sección 3.

2.1. Tareas

1. Estudiar en profundidad el problema, asegurándose de comprender bien las entradas y salidas involucradas, así como las relaciones entre éstas.

Técnica Backtracking

2. Diseñar un algoritmo basado en Backtracking para resolver el problema.
3. Diseñar varias funciones de cota.
4. 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).

5. Implementar el algoritmo y las distintas funciones de cota. Implementarlo de modo que se pueda elegir qué cota se aplica en cada ejecución concreta.
6. Realizar el análisis de la eficiencia del algoritmo en base a los distintos nodos generados, y el tiempo que se dedica a cada nodo

Técnica Branch and Bound

7. Diseñar un algoritmo basado en la técnica Branch and Bound para resolver el problema.
8. 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 del algoritmo en base a los distintos nodos generados, y el tiempo que se dedica a cada nodo

3. 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.
5. Algoritmo Backtracking. Diseño y justificación de su validez. Detalles de implementación.
6. Algoritmo Branch and Bound. Diseño y justificación de su validez. Detalles de implementación.
7. Análisis comparativo del rendimiento (en tiempo y número de nodos generados) entre las dos técnicas y las distintas funciones de cota consideradas
8. Conclusiones.

Figura 1: Estructura para entregar en prácticas

```
BK
|- Generador
|- Algoritmo
|- Instancias
|  - ni1.txt
|  - ni2.txt
|  - ...
|- Makefile
BB
| ...
```

4. Evaluación de la práctica

Esta práctica se realizará por grupos. Los alumnos habrán de entregar dos archivos en la actividad correspondiente incluida en la página de la asignatura en la plataforma Prado.

- El primer archivo será la memoria, en formato pdf.
- El segundo será el código fuente de las implementaciones realizadas empaquetado en un fichero .zip. La Figura 1 ilustra la estructura que se debe crear donde para cada problema estudiado tendremos una carpeta distinta, llamadas BK y BB. En cada carpeta el código estará organizado en tres subcarpetas de nombres: Generador, Algoritmo e Instancias. Cada una de esas carpetas incluirá todos los módulos de código fuente necesarios para generar el programa binario correspondiente las respectivas implementaciones de: generador de casos del problema y algoritmo diseñado.

Los programas aceptarán argumentos en línea de órdenes, que identificarán el caso que trata de resolver (por ejemplo, el nombre de un fichero que incluirá los datos) y enviarán a la salida estándar la solución calculada.

Además, en la carpeta raíz habrá un único fichero Makefile cuyo objetivo por defecto construirá los binarios.

La fecha límite para entregar la memoria es el día 20 de Mayo de 2024 a las 23:59 horas. Es **obligatoria** la participación de todos los miembros del grupo en la exposición.