

Máster en Sistemas Telemáticos en Informáticos

Asignatura *Optimización de Sistemas de Comunicación*

Práctica 1: Problemas e instancias

Objetivo

El objetivo de la práctica es que el alumno se familiarice con algunos problemas de optimización relacionados con los sistemas de comunicación, distinga los conceptos de problema e instancia, y sea capaz de cargar instancias para cada uno de estos problemas.

Obligatoriedad

La práctica no es obligatoria.

Prerrequisitos

El alumno debe conocer algún lenguaje de programación que le permita implementar las metodologías propuestas. El alumno dispondrá de los entornos de programación Eclipse (para programar en Java, C o C++).

MaxMin Diversity Problem

Descripción

El problema de la diversidad MaxMin (MaxMin Diversity Problem - MMDP), consiste en seleccionar un determinado número de elementos (m) de un conjunto de n elementos de tal forma que la menor de las distancias entre los elementos seleccionados sea máxima. La definición de distancia entre los elementos depende de las aplicaciones específicas. El problema MMDP se puede formular como un problema cuadrático binario:

$$\begin{aligned} \text{Maximizar } z_{MM}(x) &= \min_{x_i=x_j=1} d_{ij} \\ \text{Sujeto a } \sum_{i=1}^n x_i &= m \\ x_i &= \{0, 1\}, \quad i = 1, \dots, n. \end{aligned}$$

Instancias

Se utilizarán dos conjuntos de instancias. GKD-Ia son instancias más pequeñas (con n entre 10 y 30, y diferentes valores de m) y por tanto más sencillas de resolver. El conjunto GKD-Ic contiene instancias más grandes ($n=500$, $m=50$) que requieren un mayor esfuerzo computacional. Estos dos conjuntos de instancias junto con los mejores valores conocidos para ambos pueden descargarse desde la página web.

El formato de los ficheros es el mismo para ambos conjuntos:

- Una primera línea donde se indica el valor de n y de m
- A continuación, en cada línea se indica el índice de dos elementos y la distancias entre ellos

- Para estas instancias, existe una distancia para cada par de elementos

Ejemplo:

```
5 2
0 1 166.47234
0 2 170.18475
0 3 174.55453
0 4 153.28670
1 2 145.12186
1 3 144.42723
1 4 170.98466
2 3 176.58110
2 4 162.99632
3 4 168.48360
```

Cutwidth Problem

Descripción

Dado un grafo, el problema del minimizado del *cutwidth* (*CutWidth Problem* – CWP) consiste en encontrar una ordenación lineal del grafo de tal forma que el máximo número de aristas cortadas entre dos vértices consecutivos sea mínima. El problema del *cutwidth* se puede formular de la siguiente manera:

$$\text{Minimizar } z(x) = \max_{i \in V} c_i$$

$$c_i = |\{(j, k) \in E : x_j \leq x_i < x_k\}|$$

Donde

x es una solución al problema que consiste en una ordenación de los nodos, es decir, x_i es la posición del nodo i en la ordenación dada por la solución x

c_i es, para una determinada ordenación, el corte en el nodo i . Es decir, el número de aristas que salen de i o de nodos anteriores a i y van a parar a nodos posteriores a i .

Instancias

Se utilizará un único conjunto de instancias denominado Harwell-Boeing.

El formato de los ficheros es el siguiente:

- Una primera línea donde se indica el número de nodos (por duplicado) y el número de aristas
- El resto de líneas describen las aristas entre los nodos como dos índices que se refieren al número de nodo.

Ejemplo:

```
10 10 179
1 2
2 4
2 5
3 7
4 6
5 6
7 8
```

7	9
8	9
9	3
10	5

Capacited p-hub Problem

Descripción

En el problema del *capacited p-hub* (CPH) se tiene un determinado número de centros que pueden actuar de clientes o de servidores (también denominados *hubs*). El número de centros que pueden actuar como servidores está limitado a p , siendo el resto clientes. Cada cliente sólo se conecta a un servidor. El objetivo es determinar los p centros que actúan como servidores y conectar todos los clientes a uno de los servidores de forma que se minimice la suma de las distancias de los clientes a los servidores. Además, existe la restricción de que cada servidor sólo puede servir una determinada cantidad de recursos y cada cliente tiene una demanda que debe ser satisfecha. Las soluciones factibles al problema deben asegurarse de que todos los servidores pueden proporcionar esos recursos a sus clientes. El modelo matemático para este problema es el siguiente:

$$\begin{aligned}
 &\text{Maximizar } z(x) = \sum_j \sum_i d_{ij} x_{ij} \\
 &\text{sujeto a: } \sum_j x_{ij} = 1 \quad \forall i \\
 &\quad x_{ij} \leq y_j, \quad \forall i, \forall j \\
 &\quad \sum_j y_j = p \\
 &\quad x_{ij}, y_j \in \{0, 1\} \\
 &\quad \sum_i x_{ij} \cdot d_i \leq c \quad \forall j
 \end{aligned}$$

Donde:

x_{ij} es 1 si hay una asociación entre el cliente i y el servidor j

y_j es 1 si el nodo j es servidor, 0 si es cliente

c es la capacidad del servidor

Instancias

Se utilizarán dos conjuntos de instancias:

- phub_50_5 son instancias pequeñas, con 50 nodos de los cuales hay que escoger 5 servidores
- phub_100_10 son instancias más grandes, con 100 nodos de los cuales hay que escoger 10 servidores

El formato de los ficheros es el mismo para ambos conjuntos:

- Una primera línea donde se indica el número total de nodos, el número de nodos servidores (p) y la capacidad de cada servidor (todos los servidores tienen la misma capacidad)
- El resto de líneas describen los nodos:
 - El primer valor es el número de nodo
 - El segundo y tercer valor son las coordenadas geométricas del nodo (en un espacio

euclídeo)

- El último valor es la demanda del nodo si actúa como cliente

Ejemplo:

```
18 1043
10 2 120
1 3 60 4
2 68 49 8
3 81 53 9
4 62 94 12
5 1 57 10
6 7 98 14
7 24 94 14
8 59 95 12
9 19 13 1
10 70 86 3
```

Se pide

Las tareas a realizar durante el desarrollo de la práctica son:

- Definir las estructuras de datos adecuadas para almacenar las instancias en memoria del MMDP
- Implementar un procedimiento para cargar las instancias del MMDP en dichas estructuras de datos
- Definir las estructuras de datos adecuadas para almacenar las instancias en memoria del CWP
- Implementar un procedimiento para cargar las instancias del CWP en dichas estructuras de datos
- Definir las estructuras de datos adecuadas para almacenar las instancias en memoria del CPH
- Implementar un procedimiento para cargar las instancias del CPH en dichas estructuras de datos