

Master en Sistemas Telemáticos e Informáticos

Asignatura *Optimización de Sistemas de Comunicación*

Curso 2010/2011

Práctica 2: Creación de soluciones aleatorias

Objetivo

El objetivo de la práctica es que el alumno se familiarice con las técnicas de construcción de soluciones para los problemas planteados en la Práctica 1.

Obligatoriedad

La práctica no es obligatoria.

Prerrequisitos

El alumno debe haber realizado la Práctica 1, y en concreto, debe ser capaz de cargar instancias para los problemas planteados.

Descripción

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:

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

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 .

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:

$$\text{Maximizar } z(x) = \sum_j \sum_i d_{ij} x_{ij}$$

$$\text{sujeto a: } \sum_j x_{ij} = 1 \quad \forall i$$

$$x_{ij} \leq y_j, \quad \forall i, \forall j$$

$$\sum_j y_j = p$$

$$x_{ij}, y_j \in \{0, 1\}$$

$$\sum_i x_{ij} \cdot d_i \leq c \quad \forall j$$

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

Se pide

- Definir la estructura de una solución
- Implementar un procedimiento para calcular la función objetivo a partir de una solución al problema
- Implementar un método constructivo que genere una solución aleatoria de una instancia
- Implementar un programa que genere 1000 soluciones aleatorias por cada instancia, reportando el valor de la función objetivo y la estructura de la solución para la mejor

solución encontrada para cada instancia

Nota: cuando se utilizan números aleatorios, es útil guardar la semilla utilizada por el generador de números aleatorios para poder reproducir ciertas situaciones de interés. Por ejemplo, si se produce un error en ejecución, o se está construyendo incorrectamente una solución, guardar la semilla nos permite reproducir el error.