Práctica 3
Algoritmos Greedy

ÓSCAR BERMÚDEZ,
FRANCISCO DAVID CHARTE,
IGNACIO CORDÓN,
JOSÉ CARLOS ENTRENA,
MARIO ROMÁN
Universidad de Granada
17 de abril de 2014

Resumen

Índice

1. Terminales de venta

1.1. Implementación

Implementamos en Ruby la heurítica pedida.

```ruby
#!/usr/bin/env ruby
# encoding: utf-8

def cambio (monedas, precio)
  vuelta = []

  monedas.sort.reverse.each { |moneda|
    numero_monedas = precio / moneda
    precio = precio - numero_monedas*moneda

    vuelta.push [moneda, numero_monedas]
  }

  return vuelta
end
```

2. Red de comunicaciones

2.1. Algoritmo

Dado un conjunto de ciudades, buscamos interconectarlas con una red que minimice la longitud de red. Modelizaremos el problema como un grafo completo G del conjunto de ciudades E. Con aristas $V = \{[a,b]|\ a,b \in C\}$, donde la arista conectando los nodos a y b tiene peso igual a la distancia que los separa:

$$\forall [a,b] \in V: \quad w([a,b]) = dist(a,b) \tag{1}$$

Pretendemos interconectar ciudades de manera que la suma total de las distancias de los caminos hechos sea mínima. Es decir, buscamos el subgrafo recubridor de menor peso. Como un grafo recubridor con ciclos tiene siempre un subgrafo recubridor estrictamente contenido en él, buscamos sólo entre los grafos acíclicos. El subgrafo acíclico recubridor de menor costo es el árbol recubridor minimal.

Varios algoritmos para calcular el árbol recubridor minimal son conocidos. Ambos alcanzan una complejidad temporal de $\mathcal{O}(|E|log|V|)$ usando árboles binarios y listas de adyacencia para representar el grafo.

2.2. Triangulación de Delaunay

Para reducir la complejidad temporal del algoritmo, podemos reducir el grafo sobre que que buscamos el árbol generador minimal.

3. Segmentación de clientes

4. Asignación de trabajos

5. Asignación de aulas

6. Memorias caché

7. El problema de asignación cuadrática