

ANÁLISIS Y DISEÑO DE ALGORITMOS

Greedy method

Práctica 7 de laboratorio

Entrega: Hasta el domingo 22 de abril de 2018, 23:55h (a través de Moodle)

Asignación de coste mínimo II

Una comarca compuesta de n aldeas comunicadas mediante una red de carreteras que conforman un grafo conexo no necesariamente completo¹, está planificando la instalación de g gasolineras ($0 < g \leq n$) para dar servicio a todos sus vehículos. De cada vehículo se conoce la aldea en la que está censado y para repostar deberá ir a la población más cercana que disponga de estación de servicio. Se pretende conocer en qué aldeas habría que ubicar las gasolineras para minimizar el total de las distancias que han de recorrer todos los vehículos para ir a la gasolinera más cercana.

A diferencia del problema de la práctica anterior, en este caso puede haber más de una camino entre cualquier par de aldeas (ya sea directo o a través de otras poblaciones). Esta circunstancia hace que no se cumpla la propiedad *subestructura óptima* (Principio de Optimalidad) ya que el problema original no puede dividirse de manera eficaz en subproblemas independientes. Por lo tanto, no puede aplicarse el esquema Programación Dinámica. **En esta práctica se pide aplicar una estrategia voraz para encontrar una aproximación a la mejor ubicación posible de las gasolineras.**²

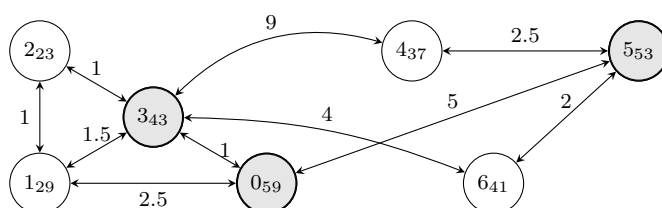


Figura 1: Posible ubicación de las tres gasolineras en los vértices sombreados en gris. Los habitantes de las aldeas 1 y 2 irían a repostar a la aldea 3 (ya que es donde está la gasolinera mas cercana), y los de las aldeas 4 y 6 irían a la 5. Los de las demás aldeas no se moverían ya que dispondrían de gasolinera en su aldea. El coste asociado a esta solución (suma total de distancias a recorrer) viene dado por la expresión $29 \cdot 1,5 + 23 \cdot 1 + 37 \cdot 2,5 + 41 \cdot 2 = 241$. **Esta ubicación, que no es la mejor, está basada en el criterio voraz de colocar las gasolineras en las ciudades con mayor censo de vehículos.**

Para realizar este ejercicio, se elegirá uno de los siguientes criterios voraces:

1. Colocar las g gasolineras en las g aldeas con mayor número de vehículos (esta es la solución mostrada en la figura 1).
2. Comenzar colocando una gasolinera en cada aldea (el coste asociado sería por lo tanto 0) y a continuación ir quitando, una a una, las $n - g$ gasolineras que hacen que el coste de la nueva configuración (con una gasolinera menos) aumente menos. Si se siguiera este criterio con el grafo de la figura 1 las gasolineras se ubicarían en las aldeas 0, 1 y 5; de lo que resulta un coste con valor 240,5.

¹En el grafo, los vértices son las aldeas y las aristas las carreteras; aunque no necesariamente el grafo dispondrá de todas sus aristas, se garantiza que todos los vértices son alcanzables entre sí (grafo conexo).

²Debe tenerse en cuenta que el método voraz no garantiza, para este problema, la solución óptima; sin embargo puede emplearse para obtener una aproximación a dicha solución, es decir, un subóptimo.

El criterio 2 produce, en general, mejores soluciones al aproximarse más al valor óptimo que para el problema del ejemplo corresponde con la ubicación 3, 4 y 5; arrojando un coste igual a 207,5.³

En cuanto a la forma de suministrar las distancias, para simplificar, el programa recibirá una matriz cuadrada d , $n \times n$, donde $d_{ii} = 0$ y $d_{ij} = d_{ji} \in R^+$ contiene la distancia del camino más corto entre la aldea i y la aldea j , $\forall i, j : 0 \leq i, j \leq n - 1$ (de esta manera no es relevante conocer cuál es la carretera que produce el camino más corto o cuántas aldeas intermedias se atraviesan). Por otra parte, el número de vehículos censados en cada población vendrá dado en un vector v con $v_i \in N$.

■ Nombre del programa, opciones y sintaxis de la orden.

El programa a realizar se debe llamar **mca_greedy**. La orden tendrá la siguiente sintaxis:

mca_greedy -f fichero_entrada

El problema a resolver se suministrará codificado en un fichero de texto cuyo nombre se recogerá a través de la opción -f. Su formato y contenido será:

- Línea 1 del fichero: Valores n y g , en este orden.
- Línea 2: Censo de vehículos: Una lista de n números enteros mayores que cero.
- Línea 3 y $n - 1$ líneas siguientes: Matriz simétrica que contiene el camino más corto entre cualquier par de aldeas según se ha descrito anteriormente.

Por tanto, el fichero contendrá $n + 2$ líneas que finalizarán con un salto de línea, salvo en todo caso, la última línea. El carácter separador entre números siempre será el espacio en blanco.

A través de *Moodle* se puede descargar un archivo comprimido con varios ejemplos y sus soluciones, entre ellos está 07a03g.p utilizado como ejemplo en este enunciado

■ Salida del programa, formato de salida y ejemplos de ejecución.

El programa mostrará dos resultados:

1. Coste asociado a la ubicación escogida, precedido de la etiqueta “**Greedy:** ”.
Por ejemplo: **Greedy:** 240.5
2. Relación de aldeas en las que se propone la instalación de gasolinera, utilizando el carácter blanco como separador y precedida de la etiqueta “**Emplacements:** ”.
Por ejemplo: **Emplacements:** 0 1 5

En estos ejemplos se ha mostrado la solución del problema de la figura 1 que corresponde al criterio voraz número 2 antes explicado. En la evaluación de esta práctica no se tendrá en cuenta qué criterio se utilice a pesar de que el número 1 es más sencillo de implementar; sin embargo, debe tenerse en cuenta que esta aproximación voraz se reutilizará en las siguientes prácticas donde el tiempo de respuesta del programa se tendrá en cuenta en la evaluación. Una implementación eficiente del criterio número 2 producirá programas más rápidos al obtener soluciones más cercanas a la óptima.

Por lo tanto, cualquiera de estas dos salidas se dará por correcta:

| | |
|---------------------------------------|---------------------------------------|
| <code>\$mca_greedy -f 07a03g.p</code> | <code>\$mca_greedy -f 07a03g.p</code> |
| <code>Greedy: 241</code> | <code>Greedy: 240.5</code> |
| <code>Emplacements: 0 3 5</code> | <code>Emplacements: 0 1 5</code> |
| <code>\$</code> | <code>\$</code> |

Es imprescindible ceñirse al formato y texto de salida mostrado, incluso en lo que se refiere a los saltos de línea o carácter separador, que en todos los casos es el espacio en blanco (aunque no hay problema si hay más de uno). La última línea debe terminar con un salto de línea

³Este valor ha sido obtenido con un método de búsqueda exhaustiva como es *Backtracking* (siguiente práctica).

(y sólo uno). **Debe respetarse los textos y formatos mostrados y en ningún caso debe añadirse texto o valores adicionales.** Deberá tratarse también posibles errores en los argumentos de la orden, no suministrar el fichero de entrada o su inexistencia, tal y como se hizo en la práctica anterior.

Normas para la entrega.

ATENCIÓN: Estas normas son de obligado cumplimiento para que esta práctica sea evaluada.

1. Se debe entregar únicamente el código fuente, con nombre `mca_greedy.cc`, y el fichero `makefile`. No hay que entregar nada más, en ningún caso se entregarán ficheros de test.
2. Es imprescindible que no presente errores ni de compilación ni de interpretación (según corresponda), en los ordenadores del laboratorio asignado.⁴ Se tratará de evitar también cualquier tipo de *warning*.
3. Todos los ficheros que se entregan deben contener el nombre del autor y su DNI (o NIE) en su primera línea (entre comentarios apropiados según el tipo de archivo).
4. Se comprimirán en un archivo `.tar.gz` cuyo nombre será el DNI del alumno, compuesto de 8 dígitos y una letra (o NIE, compuesto de una letra seguida de 7 dígitos y otra letra). Por ejemplo: `12345678A.tar.gz` o `X1234567A.tar.gz`. **Solo se admite este formato de compresión y solo es válido esta forma de nombrar el archivo.**
5. En el archivo comprimido **no debe existir subcarpetas**, es decir, al extraer sus archivos estos deben quedar guardados en la misma carpeta donde está el archivo que los contiene.
6. La práctica hay que subirla a *Moodle* respetando las fechas expuestas en el encabezado de este enunciado.

⁴Si trabajas con tu propio ordenador o con otro sistema operativo asegúrate de que este requisito se cumple.