Unit 4

Routing and Location problems

Problemas de rutas y localización

CONTENTS

1. Problema del TSP
2. Heurísticas

Al final de esta sección serás capaz de…

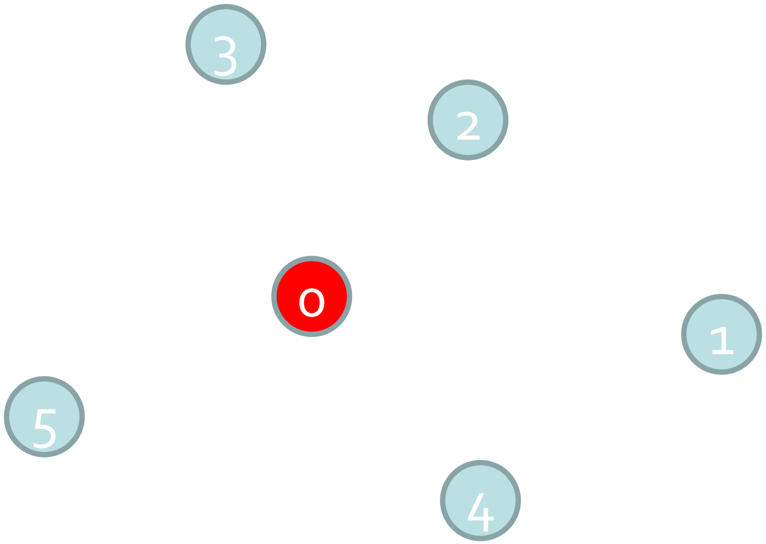
- …conocer lo que son las heurísticas para abordar problemas de optimización
- … plantear problemas en los que se quiere decidir una ruta óptima cumpliendo ciertos criterios para el TSP.
- … construir modelos de programación matemática para su resolución.
- … utilizar el Solver de Excel para la resolución de problemas de programación matemática.

TSP

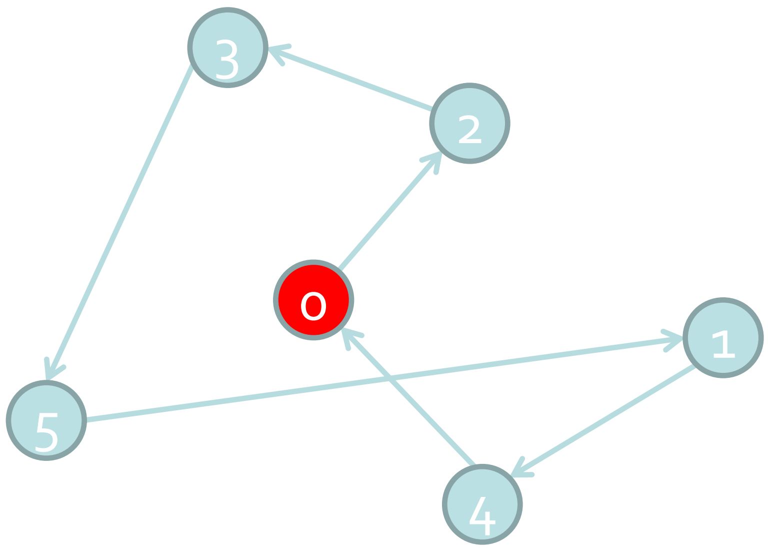
- Dado un conjunto de ciudades, encontrar la ruta que, partiendo desde una ciudad de origen, visita cada ciudad exactamente una vez, y regresa a la ciudad de origen.
- El resultado es un ciclo simple (no hay ciclos intermedios).
- Ciclo o circuito *Hamiltoniano*
- Se suele conocer como TSP (*Traveling Salesman Problem*)
- Uno de los problemas de optimización combinatoria más estudiado
- Clasificado como NP-duro (NP-hard).

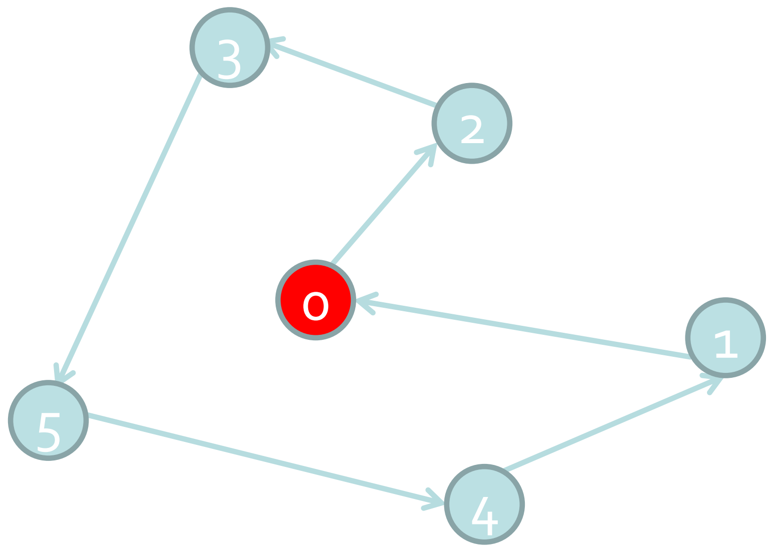
Ejemplo TSP

- Un controlador de contadores de electricidad está en el punto 0.
- Tiene que visitar 5 fábricas, situadas en los otros puntos del grafo.
- ¿Qué ruta ha de seguir para minimizar costes de transporte?



Ejemplo TSP





Definición formal del problema

- Sea $N = \{1, \dots, n\}$ el conjunto de puntos a visitar por un viajante situado en el punto 0, llamado depósito. Sea $N_0 = N \cup \{0\}$. Los nodos de N_0 se denotarán por i, j.
- Sea $A = \{(i, j): i, j \in N_0\}$ el conjunto de arcos. Sea c_{ij} el coste asociado al arco (i, j)
- (N, A) un grafo dirigido conexo.
- Encontrar la ruta que parte de 0, visita todos los nodos de N una y solo una vez, y vuelve a 0, de tal forma que el coste total de la ruta sea lo mínimo posible. Toda la ruta se realiza a traves de arcos de A.

Variables

- $x_{ij} \in \{0, 1\}$ dice si el arco (i, j) está en la ruta.
- $u_i \geq 0$ define la posición del nodo i en la ruta (número de nodos visitados antes de i).

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.:} \sum_{i \neq j} x_{ij} = 1,\;\; j = 0, ..., n. \tag{2}$$

$$\sum_{j \neq i} x_{ij} = 1,\;\; i = 0, ..., n. \tag{3}$$

$$u_i - u_j + nx_{ij} \leq n - 1, 1 \leq i \neq j \leq n \tag{4}$$

(2) Impone que de cada nodo ha de salir un arco, y (3) impone que a cada nodo ha de llegar un arco. (4) impone que no haya ciclos intermedios. Éstas son las conocidas restricciones de Miller-Tucker-Zemlin (MTZ). Existen otras restricciones de ese tipo. Este modelo no consigue resolver problemas con "muchos" nodos.

Ejemplo TSP

INSTANCIA: DATOS DE ENTRADA ¿QUÉ NECESITO?

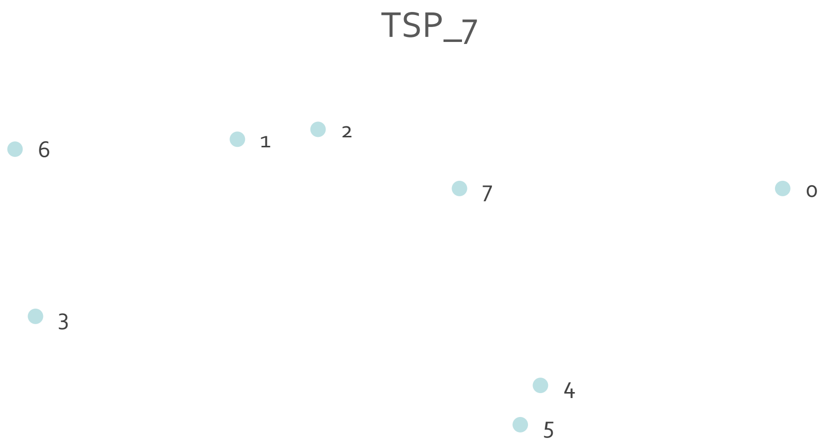
Nodo	X-coord	Y-coord
0	20	4
1	-7	9
2	-3	10
3	-17	-9
4	8	-16
5	7	-20
6	-18	8
7	4	4

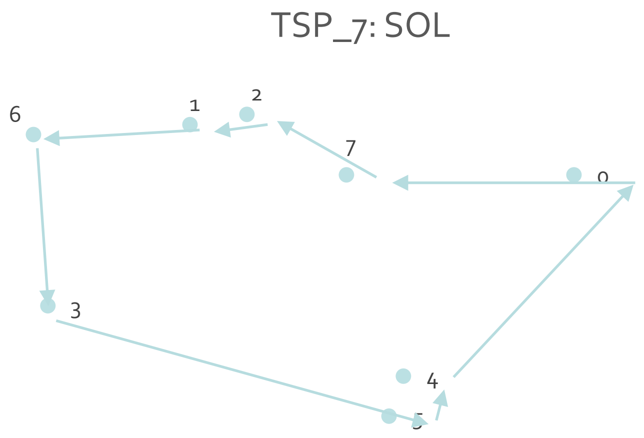
CALCULAR DISTANCIAS EUCLÍDEAS ENTRE LOS NODOS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Ejemplo TSP





Z* = 111, 265068

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Heurísticas

- Un **método heurístico** o aproximado proporciona una buena solución del problema no necesariamente óptima.
  - *"Un método heurístico es un procedimiento para resolver un problema matemático bien definido mediante una* **aproximación intuitiva***, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución"* D. de Werra y otros
  - *"Un heurístico es una técnica que busca buenas soluciones con un tiempo de computación razonable sin garantizar la optimalidad"* C.R. Reeves

Heurísticas

- Un buen algoritmo heurístico debe de tener las siguientes propiedades:
- ✓ Eficiente
- Un esfuerzo computacional realista para obtener la solución.
- ✓ Bueno (eficaz)
- La solución debe de estar, en promedio, cerca del óptimo.
- ✓ Robusto
- La probabilidad de obtener una mala solución (lejos del óptimo) debe ser baja.

Heurísticas

- Reglas de despacho: ordenaciones sencillas
- Constructivas
  - Parten de cero
  - Se va construyendo la solución
- Búsqueda local
  - Parten de una solución generada previamente
  - Mejora aplicando algún procedimiento

Heurísticas

- Constructivas
  - En cada paso añaden un elemento hasta que se completa una solución.
  - Vecino más próximo
  - Inserción
  - Ahorros
  - ...

- Búsqueda local
  - Cada solución *S* tiene un conjunto de soluciones asociadas *N(S)* conocidas como vecindario de S.
  - Cada solución *S′* del vecindario *N(S)* se obtiene aplicando un movimiento a *S.*
  - El objetivo es encontrar una solución *S′* mejor que *S* en el vecindario *N(S)*. Desde la nueva solución *S′*, se repite el procedimiento obteniendo el vecindario *N(S′)*

Búsqueda local, estrategias

- **First improvement:** el primer movimiento que nos lleve a una solución mejor, se realiza.
- **Best improvement:** se realizan todos los posibles movimientos y finalmente se realiza el movimiento con mejor evaluación de la función objetivo.
- **VNS (Variable Neighbourhood Search):** consiste en la **concatenación** de **distintos vecindarios** que se aplican de manera secuencial. Si en algún vecindario se realiza algún movimiento, la búsqueda local vuelve a comenzar desde el primer vecindario. Por ejemplo, primero se aplica el vecindario de inserción y luego el de intercambio.

Heurísticas

- Al resolver un problema de forma heurística hay que **medir** la **calidad de los resultados** ya que la optimalidad no está garantizada.
- **¿Cómo medir la calidad de un heurístico?**

➢ **Comparación con la solución óptima**

➢ **Comparación con una cota**

➢ **Comparación con otros heurísticos**

➢ Comparación con un método exacto truncado

➢ Análisis del peor caso

- **¿Cómo se compara?**
- GAP o RPD: desviación porcentual respecto a la mejor solución conocida

$$Percentage\quad Deviation = \frac{Heu_{sol} - Best_{sol}}{Best_{sol}} \cdot 100$$

- Necesitamos conjuntos de problemas o **instancias** (benchmark of instances)

- **Qué instancias utilizar?**
  - Generación propia
  - Instancias disponibles en la literatura

Generación propia

- Distintos tamaños
- Varios conjuntos de acuerdo a parámetros
- Disponibilidad para la comunidad científica
- Exhaustivo
- Adecuado para análisis estadísticos
- Discriminante

Ejemplo TSP

RUTAS POR VÉRTICES

- Recorrer los nodos de un grafo.
- Se han de satisfacer ciertas condiciones
  - Capacidades
  - Distancias entre nodos
  - ...
- **Problema del viajante**
- Problema de enrutamiento de vehículos

Heurística del vecino más próximo: ¿cómo puedo representar una solución?

Lista o vector donde la posición del nodo indica el orden en el que se visita

0	5	6	1	4	3	2	7

Heurística del vecino más próximo

1. Calcular las distancias euclídeas entre cada par de nodos
2. Para *k*=0 hasta *n-1*

   2.1 Ir al nodo más cercano al nodo k

3. Retornar la lista con el orden de visita de los nodos

En nuestro ejemplo: distancias euclídeas

Nodo	0	1	2	3	4	5	6	7
0	0,00	27,46	23,77	39,22	23,32	27,29	38,21	16,00
1	27,46	0,00	4,12	20,59	29,15	32,20	11,05	12,08
2	23,77	4,12	0,00	23,60	28,23	31,62	15,13	9,22
3	39,22	20,59	23,60	0,00	25,96	26,40	17,03	24,70
4	23,32	29,15	28,23	25,96	0,00	4,12	35,38	20,40
5	27,29	32,20	31,62	26,40	4,12	0,00	37,54	24,19
6	38,21	11,05	15,13	17,03	35,38	37,54	0,00	22,36
7	16,00	12,08	9,22	24,70	20,40	24,19	22,36	0,00

Ejemplo TSP

Solución obtenida

0	7	2	1	6	3	4	5

Valor de la Función Objetivo (VFO)

VFO = 16 + 9,22 + 4,12 + 11,05 + 17,03 + 25,96 + 4,12 + 27,29 = 114,79

Comparamos con solución óptima

GAP o **RPD**: desviación porcentual respecto a la solución óptima

$$\mathrm{RPD} = \frac{114{,}79 - 111{,}265068}{111{,}265068} \cdot 100 = 2{,}46\%$$

Ejemplo TSP

Ya tenemos una solución a partir de un heurístico constructivo

Ahora….¿Cómo podemos mejorarla?

Búsqueda local

Vecindario de inserción

Vecindario de intercambio (swap)

Ejemplo TSP

Vecindario de inserción

Partimos de una solución

Cada nodo es insertado en todas las posiciones y se deja en la posición que menor coste total obtenga

En nuestro ejemplo la solución es

0	7	2	1	6	3	4	5

Obviando el nodo 0, el nodo 7 se inserta en todas las posiciones posibles. En cada posición se evalúa (se calcula la función objetivo) y se deja donde menor coste se obtenga. Lo mismo con el resto de nodos (ciudades)

Ejemplo TSP

Vecindario de intercambio

Partimos de una solución

Cada nodo se intercambia con el resto de nodos y se realiza el intercambio que menor coste total obtenga

En nuestro ejemplo la solución es

0	7	2	1	6	3	4	5

Obviando el nodo 0, el nodo 7 se intercambia con los nodos 2, 1, 6, 3, 4 y 5. En cada intercambio se evalúa (se calcula la función objetivo) y se deja donde menor coste se obtenga. Lo mismo con el resto de nodos (ciudades)