

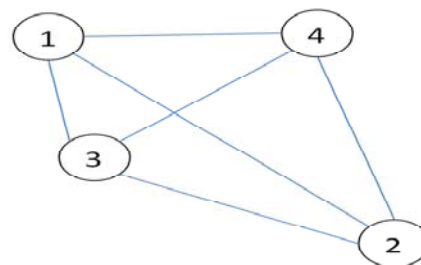
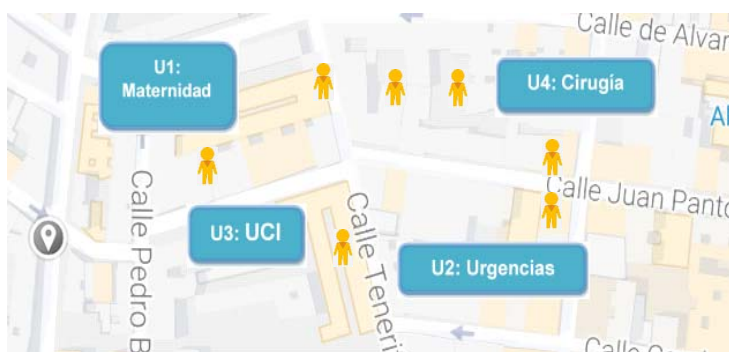
# Programación evolutiva

Facultad de Informática. Curso 2016/2017

## Práctica 2.

El objetivo de esta práctica es implementar, utilizando en lo posible el esquema, interfaz y tipos de gráficas de la práctica anterior, un algoritmo evolutivo para resolver el siguiente problema de asignación cuadrática (optimización combinatoria):

Queremos diseñar la ubicación de las diferentes  $n$  edificios de un hospital, por ejemplo: u1: Maternidad; u2: Urgencias; u3: UCI y u4: Cirugía...Queremos situarlas en  $n$  localizaciones (un ejemplo de 4):



El problema de la asignación cuadrática en este caso consiste en encontrar la asignación óptima de los  $n$  edificios del hospital en las  $n$  localizaciones disponibles, conociendo la distancia entre localizaciones y el flujo de personas por hora existente entre los edificios.

Sean  $n$  hospitales o unidades ( $u_i$ ,  $i=1,\dots,n$ ) y  $n$  localizaciones ( $l_j$ ,  $j=1,\dots,n$ ). Entonces las matrices  $F=(f_{ij})$  y  $D=(d_{kl})$ , de dimensión  $(n \times n)$  son:

- $F$  es la matriz de flujo, es decir,  $f_{ij}$  es el flujo que circula entre la unidad  $i$  a la  $j$ .
- $D$  es la matriz de distancias, es decir,  $d_{kl}$  es la distancia entre la localización  $k$  y la  $l$ .

El objetivo es minimizar el costo total, definido por:

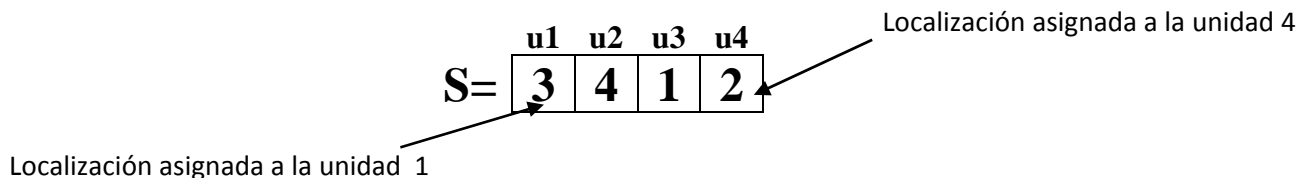
$$\min_{S \in \Pi_N} \left( \sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{S(i)S(j)} \right)$$

Donde  $\Pi_N$  es el conjunto de todas las permutaciones posibles de  $N = \{1,2,\dots,n\}$ .  $f_{ij}$  es el flujo que circula entre la unidad  $i$  a la  $j$  y  $d_{S(i)S(j)}$  se refiere a la distancia entre las localizaciones asignadas a  $i, j$  en la permutación codificada.  $F$  y  $D$  son datos de entrada. Ejemplo:

$$(F) = \begin{pmatrix} 0 & 3 & 8 & 3 \\ 3 & 0 & 2 & 4 \\ 8 & 2 & 0 & 5 \\ 3 & 4 & 5 & 0 \end{pmatrix}$$

$$(D) = \begin{pmatrix} 0 & 12 & 6 & 4 \\ 12 & 0 & 6 & 8 \\ 6 & 6 & 0 & 7 \\ 4 & 8 & 7 & 0 \end{pmatrix}$$

Se trata de buscar la solución óptima como una permutación de los valores de  $1..n$   
 $N = \{1, 2, 3, 4, \dots, n\}$ . En el ejemplo anterior una posible solución sería  $S = \{3, 4, 1, 2\}$ , cuyo coste es 358 y representa la siguiente distribución de asignaciones



El algoritmo evolutivo implementado deberá tener las siguientes características:

- Representación mediante permutaciones
- Eliminación de duplicados: se comprobará que no existan duplicados (individuos que codifiquen la misma permutación). Se reemplazarán por un individuo generado al azar o producto de una alteración del individuo a eliminar (por ejemplo, una mutación).
- Selección: Ruleta, Torneos, Estocástico, otros...
- Cruce: PMX, OX, Variantes de OX, Ciclos (CX), Recombinación de rutas (ERX), Codificación Ordinal, y algún método propio
- Mutación: Inserción, Intercambio, Inversión, Heurística y algún método propio.
- Inversión especial: operador aplicado según un porcentaje, punto inicial y punto final. El operador se aplicará siempre después del cruce y la mutación, y se reemplazará al individuo original siempre que se obtenga uno con mejor aptitud.
- Se valorará el uso de operadores distintos de los vistos en clase y que mejoren el resultado del algoritmo.
- Funciones de prueba: Se proporcionan 4 archivos de prueba para evaluar la ejecución del programa y generar el informe de resultados. Los archivos están disponibles en el campus: **ajuste.dat**, **datos12.dat**, **datos15.dat** y **datos30.dat**. El primero de ellos es para calibrar el programa y los otros 3 son los ejemplos a utilizar para el informe.
  - ajuste.dat → Coste óptimo = 50 (permutación 2 3 4 0 1)
  - datos12.dat → Coste óptimo = 224416
  - datos15.dat → Coste óptimo = 388214
  - datos30.dat → Coste óptimo = 1818146
- El formato de los archivos es el siguiente:

*tamaño de la matriz (n)*  
*matriz de distancias (dist[n][n])*  
*matriz de flujos (flujo[n][n])*

## **Documentación a entregar**

- Hay que enviar al campus virtual antes del **24 de abril a las 12:00** un archivo comprimido con el código java de la aplicación (proyecto en Eclipse o NetBeans) cuyo nombre se corresponde con el nombre del grupo y las siglas **P2**, por ejemplo **G01P2**. Es importante seguir esta notación. En el campus virtual el ejercicio está identificado como Practica 2.
- En el archivo comprimido se incluirá una breve memoria con una portada con el nombre de los integrantes del grupo y el número de grupo. La memoria deberá contener:
  - Una ejecución representativa de cada uno de los casos que se proporcionan: **datos12.dat**, **datos15.dat** y **datos30.dat**., indicando el individuo solución, aptitud máxima, mínima, media, total de cruces, total de mutaciones y total de inversiones. Gráfica de evolución.
  - Una tabla resumen con 20 ejecuciones de cada problema, incluyendo por cada ejecución la mejor solución (individuo) y su aptitud. En la parte inferior de la tabla un resumen de las 20 ejecuciones que muestre la mejor solución de esas 20 ejecuciones, la media, la peor solución y la desviación estándar.
  - En cada ejecución se usarán los mismos parámetros para las 20 ejecuciones pero diferente semilla para generar números aleatorios.
- El día de corrección será en la sesión de Laboratorio del **25 de abril** y deberán estar presentes los dos integrantes del grupo. La práctica deberán conocerla a fondo los dos componentes del grupo pues se harán preguntas a ambos indistintamente.