

# Heurísticos de Búsqueda

Tema 1 - Quadratic Assignment Problem (QAP) mediante algoritmos básicos

Josu Ceberio e Inigo Lopez-Gazpio

07/10/2024

## Índice

1. Enunciado	1
2. Código auxiliar	2
3. Entrega	2
4. Rúbrica	2

## 1. Enunciado

Una aproximación bastante directa para afrontar un problema de optimización radica en probar todas las posibles soluciones de manera masiva. Es una aproximación bastante ineficiente y naïve, aunque puede ser una solución válida para espacios de búsqueda relativamente pequeños. La mayor ventaja de este algoritmo, denominado **Brute Force**, radica en lo simple que suelen ser las implementaciones. Normalmente este algoritmo no es ampliamente utilizado ya que necesitaríamos días o incluso semanas o meses para explorar espacios de búsqueda avanzados. Este algoritmo tampoco garantiza encontrar un óptimo global, ya que nunca sabremos que este existe si no lo hemos generado.

En este ejercicio vamos a resolver una instancia del problema denominado **QAP (Quadratic Assignment Problem)**. En pocas palabras, un problema tipo QAP podría definirse como la optimización de localizaciones sobre un mapa, en el cual tenemos que definir dónde colocar las  $n$  fábricas. Las localizaciones están separadas acorde a las distancias definidas en:

$$D = [d_{ij}]_{n \times n}$$

Además, como restricción adicional tenemos que tener en cuenta que el flujo de trabajo entre las fábricas viene delimitado por:

$$H = [h_{ij}]_{n \times n}$$

por lo que nos interesa que las fábricas con un alto grado de dependencia estén situadas lo más cerca posible. Recuerda que no es posible asignar más de una fábrica a cada localización. En QAP tratamos de buscar una solución que minimice la siguiente función objetivo:

$$f(\sigma) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} h_{\sigma(i)\sigma(j)}$$

## 2. Código auxiliar

Enlace código ejemplo (jupyter-notebook).

## 3. Entrega

Entregar un documento pdf que contenga el pseudocódigo de los algoritmos empleados en el problema (RS y BF), la siguiente tabla completa y una gráfica en la que se compare la convergencia de los algoritmos acorde a la función de fitness (fitness vs steps para cada experimento).

	Tai5a.dat	Tai10a.dat	Tai20a.dat
BF (steps)			
BF (time)			
BF (best F)			
RS(100) (AVG steps)			
RS(100) (AVG time)			
RS(100) (AVG best F)			
RS(500) (AVG steps)			
RS(500) (AVG time)			
RS(500) (AVG best F)			
RS(1000) (AVG steps)			
RS(1000) (AVG time)			
RS(1000) (AVG best F)			
...			

\*Las ejecuciones del algoritmo RS se deben ponderar entre 10 repeticiones como mínimo.

## 4. Rúbrica

Esta entrega no tiene una calificación directa asociada.