

# MANUAL DE EJECUCIÓN GUIÓN 2

Grupo 5 de prácticas

Francisco Javier Sánchez Olmo

[fjso0004@red.ujaen.es](mailto:fjso0004@red.ujaen.es)

Álvaro Martin Bacas

[amb00093@red.ujaen.es](mailto:amb00093@red.ujaen.es)



Universidad de Jaén

## ÍNDICE

Descripción general del proyecto	3
Requisitos para la ejecución del proyecto	3
Configuración de parámetros	3
Ejecución del proyecto	4

# Descripción general del proyecto

En este proyecto se han desarrollado una serie de metaheurísticas con el fin de resolver un problema de TSP. Estos algoritmos son:

- Algoritmo Generacional
- Algoritmo Estacionario

## Requisitos para la ejecución del proyecto

Para poder ejecutar el proyecto, se han de tener instalados los siguientes paquetes mediante el uso del comando “pip install nombre\_del\_paquete”:

- Matplotlib (pip install matplotlib)
- Numpy (pip install numpy)
- Scipy (pip install scipy)

## Configuración de parámetros

Para que el programa lea los archivos de las ciudades bien, estos deben estar en una carpeta llamada “resources” en el directorio donde se encuentran todos los ficheros de código.

Para cada uno de los tres algoritmos, el único parámetro que necesitarán será la matriz de distancias del problema, que se construye en el propio main.

Además, internamente se asignan unos valores a las variables que dependen de los parametros indicados en el fichero config.txt (este archivo config.txt debe estar en el mismo directorio que los archivos fuente):

- **Archivos:** nombres de los archivos con las coordenadas:  
“a280.tsp”, “ch130.tsp”, “d18512.tsp”, “pr144.tsp” y “u1060.tsp”
- **Semillas:** Las semillas utilizadas son 76737859 77637859 y 56739877.
- **Algoritmos:** los algoritmos implementados en la práctica: generacional y estacionario.
- **K\_GREEDY:** número de candidatos para seleccionar mediante el greedy
- **individuos\_iniciales:** tamaño de la población inicial para los algoritmos evolutivos.
- **numero\_max\_elites:** número de individuos que se mantendrán con cada iteración para conservar el elitismo.
- **k\_best\_gen y k\_best\_est:** número de individuos que serán seleccionados para el torneo de ganadores.

- **k\_worst\_gen** y **k\_worst\_est**: número de individuos que serán seleccionados para el torneo de perdedores.
- **Parada\_iteraciones** y **parada\_tiempo**: condiciones de parada. O bien el número máximo de evaluaciones, o bien por tiempo de ejecución.
- **Porcent\_cruce\_gen**: Probabilidad de usar el operador de cruce del algoritmo generacional.
- **Porcent\_mutacion**: Probabilidad que tiene un hijo de mutar.
- **Individuos\_iniciales\_aleatorios**: Porcentaje de población completamente aleatoria.
- **Individuos\_iniciales\_greedy**: Porcentaje de población creada mediante el algoritmo greedy.
- **Tipo\_cruce**: para seleccionar entre los operadores de cruce “OX2” y “MOC”.

## Ejecución del proyecto

Se podrá ejecutar desde el IDE o desde la consola el archivo main.py.

Este archivo ejecutará los algoritmos indicados en todas sus diferentes versiones, para los archivos indicados, y con las semillas indicadas también en config.txt

Se mostrará por consola la solución obtenida por cada versión del algoritmo, junto con los parámetros de dicha ejecución, además del tiempo que tarda en ejecutar. Para más detalles, se puede consultar el fichero registro.log que se irá generando e incluirá información acerca de cómo se va comportando el algoritmo.