

|  |  |
| --- | --- |
| Algoritmos metaheurísticos  Implementación, resultados y análisis |  |
| Moreno Cabello, César  Sánchez Nájera, Elena  Grado en ingeniería Informática  Universidad de Jaén  Curso 2025/2026 |  |

Índice

[Descripción del problema 3](#_Toc211175986)

[Parámetros de configuración 3](#_Toc211175987)

[Estructura del proyecto 3](#_Toc211175988)

[Carpeta principal 3](#_Toc211175989)

[Carpeta “datos” 3](#_Toc211175990)

[Carpeta “clases” 3](#_Toc211175991)

[Carpeta “logs” 4](#_Toc211175992)

[Carpeta “modulos” 4](#_Toc211175993)

Descripción del problema

Para el análisis y comparación de los diferentes algoritmos metaheurísticos, se utilizará un problema común, del que se estudiarán cuatro instancias diferentes.

Dicho problema recibe el nombre de **“Cadena de montaje”**. Se trata de mejorar el flujo de piezas entre los diferentes departamentos dentro de la fábrica FORD de Valencia. En cada instancia, los datos de partida son dos matrices cuadradas de igual tamaño: F, definida como la matriz de flujo de piezas que pasan entre departamentos; y D: la matriz de distancias entre departamentos.

La solución vendrá representada como una permutación de números naturales, donde la posición indica el departamento y el contenido la localización. El objetivo del problema es minimizar el coste de asignación de los departamentos para las diferentes localizaciones.

Parámetros de configuración

Con objetivo de agilizar las ejecuciones, se ha creado un fichero *config.txt*, donde se indicarán los parámetros que se han de tener en cuenta para una determinada ejecución. El uso de este fichero permitirá implementar varios algoritmos para diferentes datos y con múltiples parámetros en una sola ejecución. El formato que debe seguir el fichero de configuración debe ajustarse al siguiente:

* DATA = ./datos/*NombreArchivoDatos Ruta del archivo de datos*
* ALG = *NombreAlgoritmo Nombre del algoritmo*
* Seed = *Semilla Semilla a utilizar en algoritmos aleatorios (opcional)*
* K = *n Parámetro de aleatoriedad en greedy\_aleatorizado (opcional)*
* Max\_iteraciones = *n Número máximo de iteraciones para parar un algoritmo (opcional)*
* Tenencia = *n Tenencia tabú (opcional)*
* Oscilación = *n Oscilación tabú (opcional)*
* Estancamiento = *n Estancamiento tabú (opcional)*

Los parámetros de configuración opcionales podrán inicializarse a 0 si no se desean utilizar, pero debe **mantenerse la estructura** general del archivo para su correcto funcionamiento.

Además de cumplir con el formato del archivo de configuración, se ha de tener en cuenta que los archivos de datos con los que se trabajará deberán estar incluidos en la carpeta del proyecto, “**datos**”. Como es evidente, el programa solo funcionará si recibe como argumento el fichero de configuración, que como **mínimo** deberá contener la dirección de un **archivo de datos** y el nombre de un **algoritmo** para trabajar.

Estructura del proyecto

El proyecto se divide en carpetas para facilitar la clasificación de los ficheros. Así pues, se encuentra el directorio dividido entre las siguientes carpetas:

### Carpeta principal

Contiene el fichero de configuración, así como el *main* y el *README*.

### Carpeta “datos”

En este directorio, el usuario deberá introducir los ficheros de datos con los que desea trabajar, ya que el programa los buscará ahí en las ejecuciones donde sea necesario.

### Carpeta “clases”

Aquí se definen clases auxiliares, como la clase *configurador*, *extractor*, o *logs*. Las dos primeras clases son útiles para importar la información del fichero de configuración en el programa. Mientras tanto, la clase *logs* servirá para extraer los datos obtenidos en cada ejecución de cada algoritmo a un fichero de texto, donde el usuario podrá consultar el comportamiento de las metaheurísticas de manera individualizada y sin perder los datos tras finalizar el programa.

### Carpeta “logs”

En “**logs**” se guardarán los ficheros que resumen el comportamiento de los diferentes algoritmos. Dentro de los archivos se incluirá la configuración aplicada, con objeto de tener claro qué pasó durante la ejecución, así como la asignación de la permutación, el costo que supone y el tiempo de ejecución del algoritmo. Cada documento será nombrado la estructura “*NombreAlgoritmo*\_ *NombreArchivoDatos*\_ *Semilla*”, donde semilla será un parámetro opcional en función del algoritmo que se haya definido.

### Carpeta “modulos”

Por último, se encuentran archivos que contienen el funcionamiento general del programa, como *heurísticas, donde se define cómo se implementa cada algoritmo, func\_auxiliares*, donde se definen funciones de ayuda, como el cálculo del costo o la factorización; o *prints*, donde hay funciones que sirven de ayuda a la hora de imprimir resultados o errores por pantalla.