

**Grado en Ingeniería Informática**

**Metaheurísticas**

**Curso 2019/2020**



**Universidad de Jaén**

**Práctica 3**

# 1. Objetivos

El objetivo de esta práctica es estudiar el funcionamiento de las Metaheurísticas híbridas, y en concreto de los Algoritmos Meméticos. Para ello, se requerirá que el estudiante implemente un:

- Algoritmo Memético Estacionario.

La fecha límite de entrega será el **martes 17 de diciembre de 2019 antes de las 23:59 horas**. La entrega se realizará a través de la plataforma virtual.

# 2. Trabajo a realizar

Este guion será una continuación del segundo, por tanto, el trabajo a realizar y los casos de estudio permanecen sin cambios.

# 3. Algoritmo memético estacionario

Se aplicará un único memético con un modelo evolutivo estacionario que parte del mismo algoritmo presentado en la Práctica 2 pero con la **modificación del esquema de evolución**:

- Pasamos a un esquema estacionario donde sólo evolucionan (entre generaciones) dos individuos de la población.
- El reemplazamiento de estos dos individuos se hace con respecto a los dos peores de la población de la anterior generación.
- Desaparece el concepto de elitismo en este algoritmo.

## Enfoque estacionario

El resto de componentes del algoritmo que se mantienen son:

- Operador de selección basada en un torneo binario, es decir con  $k=2$ , aplicándose tantas veces como individuos tengamos en la población. En este proceso todos los individuos de la población inicial (incluyendo los

individuos ganadores en la población anterior) entran en el proceso del Torneo.

- Operador de cruce se deberán implementar dos operadores el cruce OX2 y el cruce MOC.
- Operador de mutación será un intercambio de tres posiciones de la solución.

## Memético

El algoritmo ahora incorpora la Búsqueda Tabú (implementado en la Práctica 1) en la etapa posterior a la recombinación de individuos del algoritmo estacionario para convertirse en un algoritmo memético, tal y como hemos visto en clase de teoría.

Se va a aplicar la Búsqueda Tabú sobre los dos individuos que estamos evolucionando, una vez hayan sido cruzados y/o mutados. Este proceso se aplicará cada 50 generaciones. Además, vamos a modificar la Búsqueda Tabú de forma que se reducirá el número de iteraciones de este algoritmo (según se indique por parámetro), y también se eliminará la oscilación estratégica por lo que la memoria a largo plazo carecerá de valor.

## Parámetros de los algoritmos

Se detallan a continuación todos los parámetros de los algoritmos:

- Tamaño de la población de 25 individuos.
- Máximo número de generaciones 1000 (condición de parada).
- Probabilidad de cruce de 1.0.
- Probabilidad de mutación para cada gen será de 0.05.
- Generaciones sobre las que se aplica la Tabú: 50.
- Número de iteraciones de la Tabú: 100 y 500.

## Resumen de los algoritmos

- Memético estacionario con cruce OX2 con 100 iteraciones de Tabú.
- Memético estacionario con cruce OX2 con 500 iteraciones de Tabú.
- Memético estacionario con cruce MOC con 100 iteraciones de Tabú.
- Memético estacionario con cruce MOC con 500 iteraciones de Tabú.

## 4. Tablas de resultados y análisis

Seguiremos trabajando el mismo formato que en el guión anterior.

Para ello, se cogerán los resultados obtenidos en la Práctica 2 y se incluirán los mismos obtenidos por los algoritmos diseñados en este guión.

El análisis que se debe realizar de hacerse de la siguiente forma y justificarlo:

1. Algoritmo memético OX2 con 100 iteraciones Vs. 500 iteraciones.
2. Algoritmo memético MOC con 100 iteraciones Vs. 500 iteraciones.
3. El mejor de 1) Vs. el mejor de 2)

## 5. Documentación y normativa

En este caso también seguiremos con la misma normativa que ya hemos trabajado en la Práctica 1:

- Solo se admitirá el formato **ZIP**. No se corregirán guiones en cualquier otro formato al indicado.
- La memoria que acompañará en la entrega será exclusivamente en formato PDF y se incluirá una portada con:
  - o Identificación de los dos alumnos (Nombre, apellidos y DNI).
  - o Identificación del guión.
  - o Identificación del grupo de los alumnos.
  - o Algoritmos implementados en la práctica.
- Los nombres de los ficheros siguen el mismo formato anterior junto con el código que se detalla más adelante donde:
- Una breve descripción.
- Descripción en **pseudocódigo** de las **estructuras de los genéticos** y de todas aquellas **operaciones relevantes** de cada algoritmo. El pseudocódigo **deberá forzosamente reflejar la implementación y el desarrollo realizados incluyendo los diseños y representación de los modelos evolutivos**, y no ser una descripción genérica extraída de las transparencias de clase o de cualquier otra fuente. La descripción de cada algoritmo no deberá ocupar más de **2 páginas**.

- Breve explicación del **procedimiento considerado para desarrollar la práctica**: implementación a partir del código proporcionado en prácticas o a partir de cualquier otro, o uso de un framework de metaheurísticas concreto. Inclusión de un pequeño **manual de usuario describiendo el proceso para que el profesor de prácticas pueda replicarlo**.
- Experimentos y análisis de resultados:
  - o Descripción de los valores de los parámetros considerados en las ejecuciones de cada algoritmo (**incluyendo las semillas utilizadas**).
  - o Resultados obtenidos según el formato especificado.
  - o Análisis de resultados. El análisis deberá estar orientado a **justificar** (según el comportamiento de cada algoritmo) **los resultados** obtenidos en lugar de realizar una mera “lectura” de las tablas. Se valorará la inclusión de otros elementos de comparación tales como gráficas de convergencia, boxplots, análisis comparativo de las soluciones obtenidas, representación gráfica de las soluciones, etc.
  - o Análisis de tiempos y desviaciones, e influencia del tamaño de la instancia en los resultados obtenidos, etc.
  - o Por último, es clave obtener un fichero de registro que vaya almacenando la información de la ejecución del algoritmo indicando las soluciones, en qué generación fueron creadas y costes que va obteniendo en las distintas iteraciones los mejores de cada generación.
- Referencias bibliográficas u otro tipo de material distinto del proporcionado en la asignatura que se haya consultado para realizar la práctica (en caso de haberlo hecho).

Aunque lo esencial es el contenido, también debe cuidarse la presentación y la redacción. **La documentación nunca deberá incluir listado total o parcial del código fuente.**

En lo referente al **desarrollo de la práctica**, se entregará una carpeta llamada **software** que contenga una versión ejecutable de los programas desarrollados, así como los ficheros de datos de los casos del problema y el código fuente implementado o los ficheros de configuración del framework empleado. El código fuente o los ficheros de configuración se organizarán en la estructura de directorios que sea necesaria y deberán colgar del directorio FUENTES en el raíz. Junto con el código fuente, hay que incluir los ficheros necesarios para construir los ejecutables según el entorno de desarrollo empleado (tales como \*.prj, makefile, \*.ide, etc.). La versión ejecutable de los programas y los ficheros de datos se incluirán en un subdirectorio del raíz de nombre BIN. En este mismo directorio se adjuntará un pequeño fichero de texto de nombre LEEME que contendrá breves reseñas sobre cada fichero incluido en el directorio. Es

importante que los programas realizados puedan leer los valores de los parámetros de los algoritmos desde fichero, es decir, que no tengan que ser recompilados para cambiar éstos ante una nueva ejecución. Por ejemplo, la semilla que inicializa la secuencia pseudoaleatoria debería poder especificarse como un parámetro más.

La entrega se llevará a cabo a través de la actividad correspondiente en ILIAS. El plazo de entrega termina el 17 de diciembre de 2019 a las 23:59.

**En caso de que el comportamiento del algoritmo en la versión implementada/desarrollada no coincida con la descripción en pseudocódigo o no incorpore las componentes requeridas, se podría reducir hasta en un 50% la calificación del algoritmo correspondiente.**