



## PRÁCTICA 3

*Sistema de Colonias de Hormigas*

Autores:

David Díaz Jiménez 77356084T

Andrés Rojas Ortega 77382127F

Grupo 2

# Metaheurísticas

## Informe de prácticas

David Díaz Jiménez, Andrés Rojas Ortega

### Contents

<b>1</b>	<b>Definición y análisis del problema</b>	<b>2</b>
1.1	Representación de la solución . . . . .	2
1.2	Función objetivo . . . . .	2
1.3	Operadores comunes . . . . .	2
<b>2</b>	<b>Clases auxiliares</b>	<b>2</b>
2.1	Archivo . . . . .	2
2.2	Configurador . . . . .	3
2.3	GestorLog . . . . .	3
2.4	Metaheurísticas . . . . .	3
2.5	Pair . . . . .	3
2.6	RandomP . . . . .	3
2.7	Timer . . . . .	3
<b>3</b>	<b>Pseudocódigo</b>	<b>4</b>
3.1	algoritmo principal . . . . .	4
<b>4</b>	<b>Experimentos y análisis de resultados</b>	<b>4</b>
4.1	Procedimiento de desarrollo de la práctica . . . . .	4
4.1.1	Equipo de pruebas . . . . .	4
4.1.2	Manual de usuario . . . . .	5
4.2	Parámetros de los algoritmos . . . . .	5
4.2.1	Sistema de Colonias de Hormigas . . . . .	5
4.2.2	Semillas . . . . .	5
4.3	Análisis de los resultados . . . . .	5

## 1 Definición y análisis del problema

Dado un conjunto  $N$  de tamaño  $n$ , se pide encontrar un subconjunto  $M$  de tamaño  $m$ , que maximice la función:

$$C(M) = \sum_{s_i, s_j \in M} d_{ij}$$

donde  $d_{ij}$  es la diversidad del elemento  $s_i$  respecto al elemento  $s_j$

### 1.1 Representación de la solución

Para representar la solución se ha optado por el uso de un vector de enteros, en el que el elemento contenido en cada posición se corresponde con un integrante de la solución. La solución vendrá dada por las siguientes restricciones:

- La solución no puede contener elementos repetidos.
- Debe tener exactamente  $m$  elementos.
- El orden de los elementos es irrelevante.

### 1.2 Función objetivo

$$C(M) = \sum_{s_i, s_j \in M} d_{ij}$$

### 1.3 Operadores comunes

El operador de intercambio es el 1-opt, se seleccionara un elemento de la solución actual en base a un criterio y se sustituirá por un elemento que no pertenece a la solución.

## 2 Clases auxiliares

A continuación se enumeran las diferentes clases auxiliares utilizadas en el programa acompañadas de una breve descripción de las mismas.

**nota:** Para obtener información detallada se deben consultar los comentarios insertados en el código de cada una de las clases.

### 2.1 Archivo

Esta clase se encarga de almacenar toda la información que se encuentra dentro de cada archivo que contiene cada uno de los problemas.

## **2.2 Configurador**

Utilizamos esta clase para leer y almacenar los parámetros del programa que se encuentran dentro del archivo de configuración.

## **2.3 GestorLog**

La función principal de esta clase es la administración de los archivos Log del programa y el almacenamiento de información para debug en los mismos.

## **2.4 Metaheurísticas**

Esta clase se utiliza para lanzar la ejecución de los algoritmos para cada problema facilitado como parámetro.

## **2.5 Pair**

Representa un par formado por un candidato y un coste asociado a este.

## **2.6 RandomP**

Clase para generar números aleatorios.

## **2.7 Timer**

Clase para gestionar los tiempos de ejecución del algoritmo.

## 3 Pseudocódigo

### 3.1 algoritmo principal

---

**Algorithm 1** Sistema Colonia Hormigas

---

```
matrizFeromonas  $\leftarrow$  InicializarFeromona()  
while iteracionesRealizadas  $\leq$  limiteIteraciones  $\wedge$  tiempoEjecucion  $\leq$  600  
do  
  colonia  $\leftarrow$  InicializarColonia()  
  for i=1; i<tamañoHormiga; i++ do  
    for hormiga  $\in$  colonia do  
      lrc  $\leftarrow$  GenerarLRC(hormiga)  
      AplicarReglaTransicion(lrc, hormiga)  
    end for  
    ActualizarFeromonaLocal()  
  end for  
  TareasDemonio()  
   $\emptyset \leftarrow$  colonia  
  iteracionesRealizadas  $\leftarrow$  iteracionesRealizadas + 1  
end while
```

---

## 4 Experimentos y análisis de resultados

### 4.1 Procedimiento de desarrollo de la práctica

Para realizar la práctica, se ha optado por implementar las heurísticas propuestas en el lenguaje de programación JAVA. El ejecutable que se entrega junto a este documento ha sido compilado bajo APACHE NETBEANSIDE 12.0.

#### 4.1.1 Equipo de pruebas

Los resultados de las heurísticas han sido obtenidos en el siguiente equipo:

- Host:
- S.O:
- Kernel:
- CPU:
- GPU:
- GPU:
- Memoria RAM :

### 4.1.2 Manual de usuario

Para ejecutar el software asegúrese de que el archivo .jar proporcionado se ubica en el mismo directorio que la carpeta *archivos*.

Cuando se muestre la GUI, podrá seleccionar la heurística que desee mediante el botón correspondiente. Una vez empiece la ejecución de una heurística no será posible seleccionar otra hasta que finalice su ejecución. Los resultados finales se mostrarán en el cuadro de texto, a su vez, se generan los log correspondientes a cada archivo y semilla en la carpeta Log.



Figure 1: GUI

## 4.2 Parámetros de los algoritmos

### 4.2.1 Sistema de Colonias de Hormigas

### 4.2.2 Semillas

Para la generación de números pseudoaleatorios se utiliza una semilla previamente definida en el archivo de configuración, en este caso es 77356084. Esta semilla se va rotando en las 5 iteraciones de cada archivo.

77356084  $\rightarrow$  73560847  $\rightarrow$  35608477 ...

## 4.3 Análisis de los resultados

## References

- [1] Umbarkar, Dr. Anantkumar & Sheth, P.. (2015). CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW. ICTACT Journal on Soft Computing ( Volume: 6 , Issue: 1 ). 6. 10.21917/ijsc.2015.0150.
- [2] <https://sci2s.ugr.es/graduateCourses/Metaheuristics>
- [3] <https://sci2s.ugr.es/graduateCourses/Algoritmica>