

A dark blue vertical bar is on the left. A blue arrow points right from it, containing the date.

13-6-2022

# PRÁCTICA 3: ALGORITMOS DE OPTIMIZACIÓN BASADOS EN COLONIAS DE HORMIGAS

Several thin, curved lines in shades of blue and grey sweep upwards from the bottom left corner.

José M<sup>a</sup> Prieto Crespo  
UNIVERSIDAD DE HUELVA

## Contenido

Introducción.....	2
SH.....	3
SHE .....	5
SCH .....	7
Comparación.....	9

## Introducción

Práctica cuyo objetivo es estudiar el funcionamiento de los Algoritmos de Optimización Basados en Colonias de Hormigas, para la que se implementan el Sistema de Hormigas (**SH**), el Sistema de Hormigas Elitista (**SHE**) y el Sistema de Colonias de Hormigas (**SCH**). Estas variantes del algoritmo se usarán para resolver el **problema del Viajante del Comercio**.

Para ello contamos con dos data sets, uno que cuenta con 130 ciudades y otro con 280. En la lectura de cada uno de los data sets solo he tenido en cuenta las coordenadas de cada ciudad, ya que los índices puedo tomarlos del array donde las guarde.

El proyecto está dividido en 4 scripts:

- **fAux:** cuenta con las funciones que van a usar todo el resto de scripts, como dibujar los caminos, leer las ciudades o la función de coste.
- **SH:** implementación del sistema de hormigas.
- **SHE:** implementación del sistema de hormigas elitista.
- **SCH:** implementación del sistema de colonias hormigas.

Coste greedy para:

- **Ch130:** 7595 kms
- **A280:** 3125 kms

Solución óptima para:

- **Ch130:** 6110kms
- **A280:** 2579 kms

## SH

**Número de hormigas:** 10.

**Criterio de parada:** 10000 \* n o 5min máx. de ejecución.

**Alpha:** 1.

**Beta:** 2

**Parámetro evaporación:** 0.1

**Semillas probadas:** [10, 20 ,30]

Para optimizar cálculos cuento con varias matrices (estas matrices las heredarán el resto de algoritmos):

- **L:** array formada por “m” filas y 131/281 columnas (esa última columna es para incluir al nodo inicial), de forma que permita contener los distintos caminos formados por cada hormiga.
- **LnV:** array con “m” filas y 130/280 columnas, donde las ciudades ya visitadas aparecen con un -1.
- **D:** matriz de distancias de dimensiones 130x130 o 280x280.
- **nH:** matriz de heurística calculada a partir de D y con sus mismas dimensiones.

Ejecución para ch130:

### Evaluaciones:

13920	14450	14040
-------	-------	-------

### Costes:

6575	6575	6575
------	------	------

### Evaluaciones hasta el óptimo:

7860	7860	7860
------	------	------

Ejecución para a280:

### Evaluaciones:

3110	3170	3050
------	------	------

### Costes:

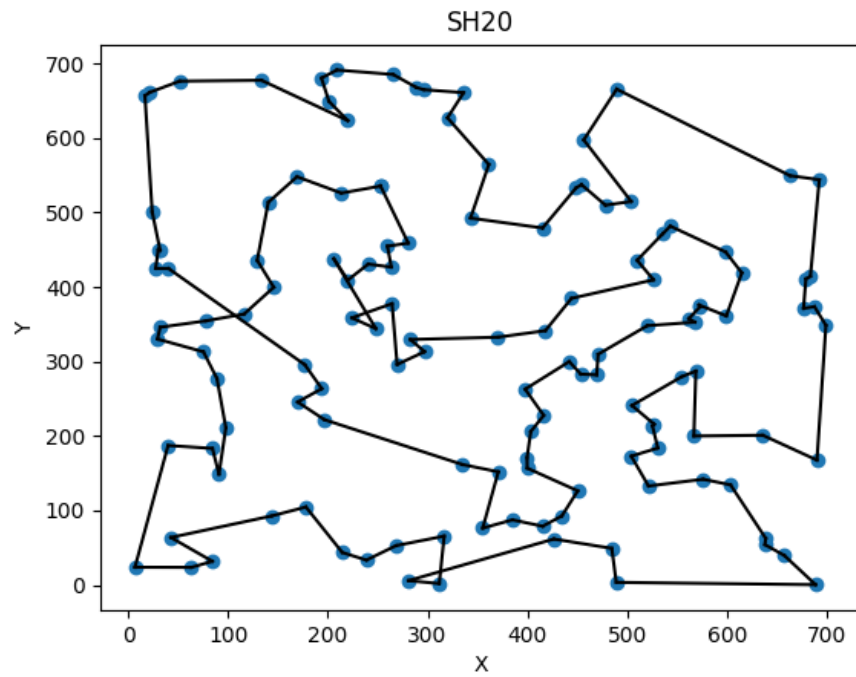
3365	3365	3365
------	------	------

### Evaluaciones hasta el óptimo:

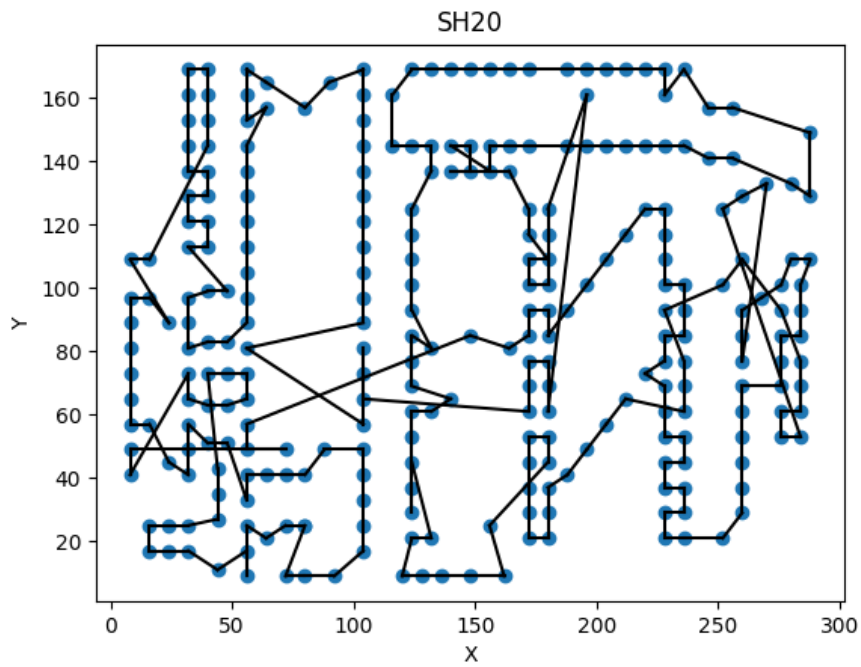
1940	1940	1940
------	------	------

## CAMINOS PARA CADA DATASET

### CH130



### A280



## SHE

Mismos parámetros que el SH, solo que añadimos el número de hormigas elitista (e) para la actualización de las feromonas pertenecientes a la mejor solución global.

**Número de hormigas elitistas:** 15

Ejecución para ch130:

### Evaluaciones:

14080	14310	14130
-------	-------	-------

### Costes:

6706	6706	6706
------	------	------

### Evaluaciones hasta el óptimo:

1510	1510	1510
------	------	------

Ejecución para a280:

### Evaluaciones:

3150	3170	3070
------	------	------

### Costes:

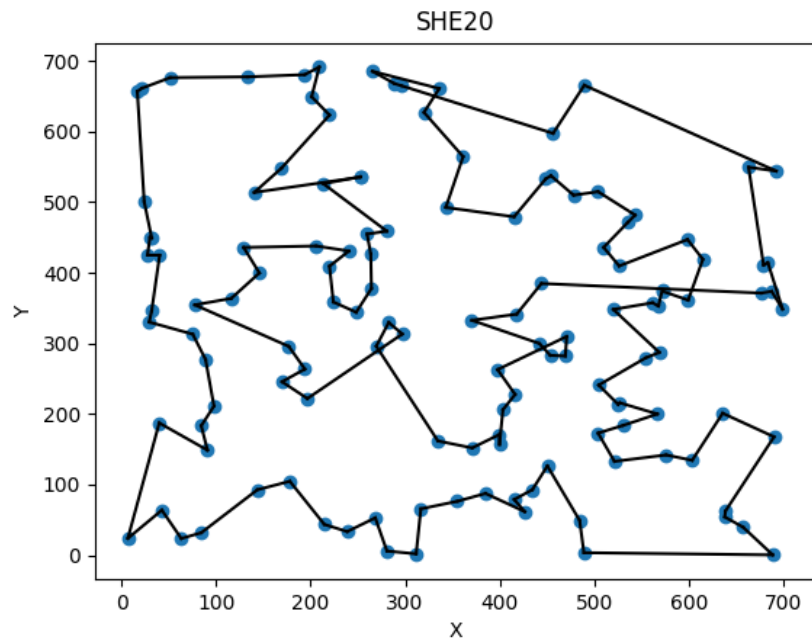
3152	3152	3152
------	------	------

### Evaluaciones hasta el óptimo:

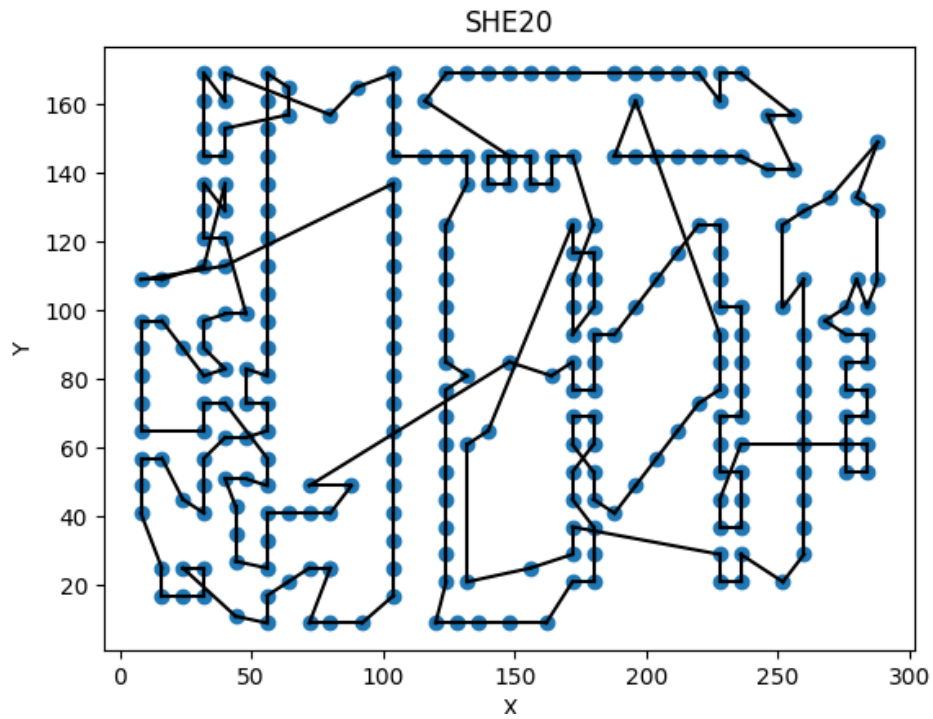
2910	2910	2910
------	------	------

## CAMINOS PARA CADA DATASET

### CH130



### A280



## SCH

A partir del SH, tenemos que añadir la actualización local de las feromonas así como la actualización global, además de modificar la regla de transición.

**Parámetro para la actualización local(fi): 0.1**

**Parámetro para la regla de transición(q0): 0.98.** Debido a este parámetro, el 98% de las veces va a realizar la regla de transición exclusiva de la colonia, en caso contrario realiza la del sistema de hormigas normal.

Ejecución para ch130:

### Evaluaciones:

29370	29410	29240
-------	-------	-------

### Costes:

6796	6796	6796
------	------	------

### Evaluaciones hasta el óptimo:

7520	7520	7520
------	------	------

Ejecución para a280:

### Evaluaciones:

6940	6890	6800
------	------	------

### Costes:

2956	2956	2956
------	------	------

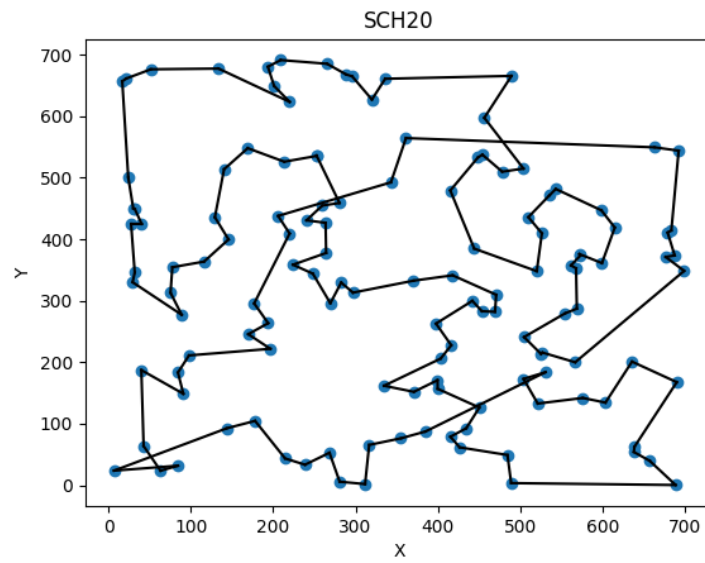
### Evaluaciones hasta el óptimo:

250	250	250
-----	-----	-----

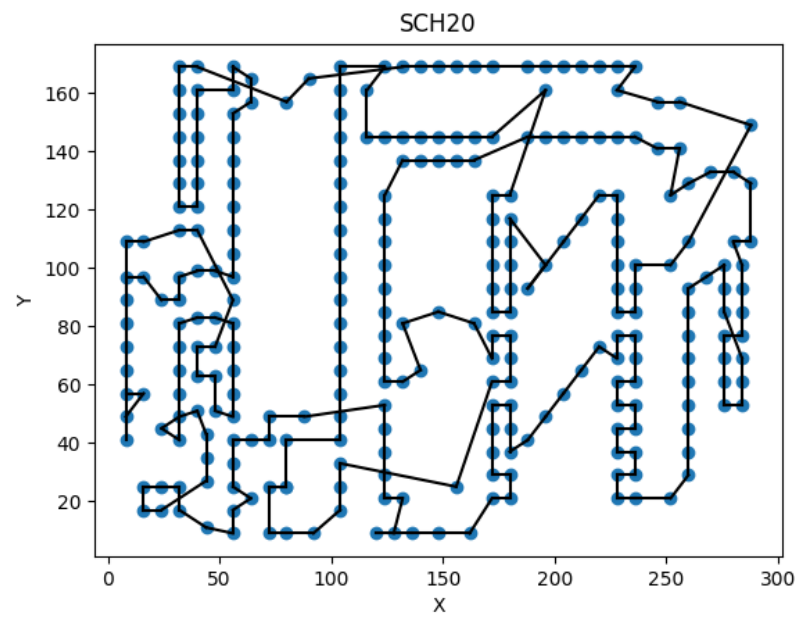


## CAMINOS PARA CADA DATASET

### CH130



### A280



## Comparación

Para CH130:

Algoritmo	Ev. Medias	Ev. Mejor	Desv. Ev	Mejor Kms	Media Kms	Desv. Kms
SH	14136.66	13920	226.91	6575	6575	0
SHE	14173.33	14080	98.77	6706	6706	0
SCH	29340	29240	72.57	6796	6796	0
Greedy	1	1	0	7595	7595	0

A pesar de que el **SH** me da mejores resultados que el **SHE**, para el CH130 el **SHE** obtiene la solución óptima 7 veces antes, por lo tanto en términos de eficiencia para este caso es mejor el **SHE**, aunque en cuanto al mejor resultado obtenido ha sido por parte del **SH**, superando incluso al **SCH**, tanto por los kilómetros como por las evaluaciones totales.

Para A280:

Algoritmo	Ev. Medias	Ev. Mejor	Desv. Ev	Mejor Kms	Media Kms	Desv. Kms
SH	3110	3050	48.98	3365	3365	0
SHE	3130	3070	43.20	3152	3152	0
SCH	6876.66	6800	57.92	2956	2956	0
Greedy	1	1	0	3125	3125	0

Para este segundo dataset se obtiene lo esperado, es decir, que el **SCH** sea el que mejores resultados me de en términos de kms, superando por casi 200 kms al **Greedy**. Además, aunque sea el que más evaluaciones medias haga, solo tarda 250 iteraciones en obtener la solución óptima.

En ambos casos la desviación de los kms me da 0 ya que para todas las semillas el valor inicial es el mismo, por lo que encuentra siempre la mejor solución para cada algoritmo con dicho valor inicial. Una de las mejoras sería cambiar ese valor inicial, ya que permitiría explorar un poco más el hecho de que el valor inicial (y por ende, el final) fuese diferente para cada semilla.