

Modelos Bioinspirados y Heurísticas de Búsqueda

Práctica 3

Daniel Pérez Rodríguez

Índice

Introducción.....	3
Algoritmos.....	3
Clase Hormiga.....	3
Greedy.....	3
Sistema de Hormigas (SH).....	4
Sistema de Hormigas Elitistas (SHE).....	6
Sistema de Colonias de Hormigas (SCH).....	7
Comparación de resultados.....	8
Como ejecutar la práctica.....	9

Introducción

El objetivo de la práctica es estudiar y desarrollar los siguientes algoritmos:

- Algoritmo de Sistema de Hormigas (SH)
- Algoritmo de Sistema de Hormigas Elitistas (SHE)
- Algoritmo de Sistema de Colonias de Hormigas (SCH)

El código desarrollado deberá encontrar el camino más corto posible para recorrer un número de ciudades determinadas. Para ello emplearemos los ficheros **ch130.tsp** y **a280.tsp**. Además, estos algoritmos deberán ser comparados con un algoritmo greedy. Se han utilizado tres semillas para las ejecuciones finales.

Algoritmos

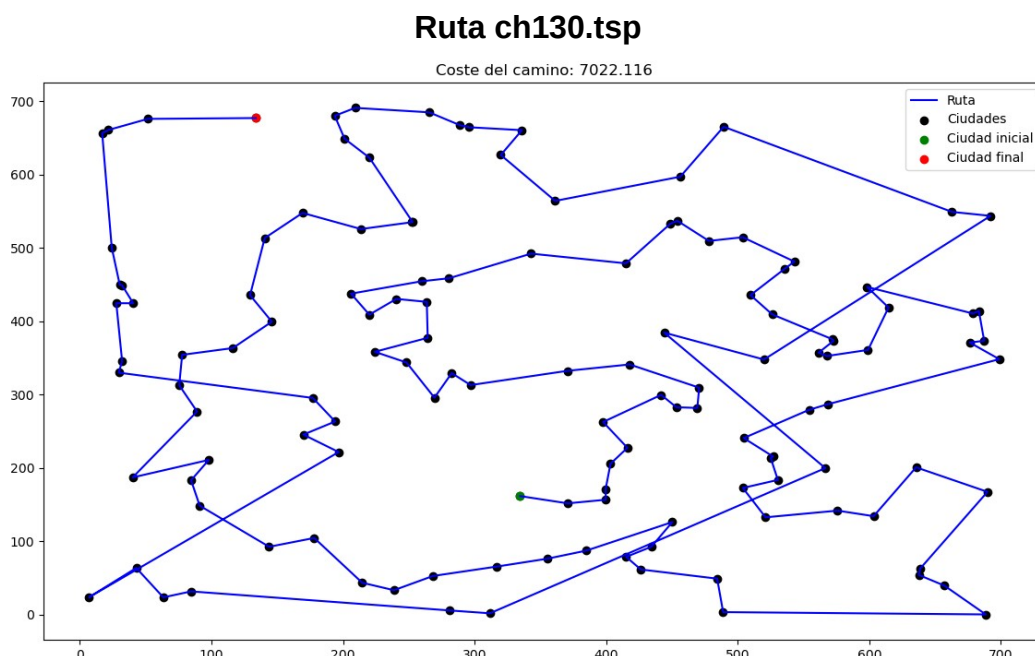
Para la implementación de los algoritmos se ha utilizado una estructura tipo grafo. En la misma, cada nodo representa una ciudad y almacena las coordenadas de la misma así como una lista con los nodos más cercanos a cada ciudad en cuestión. Todos los nodos están interconectados entre ellos mediante aristas. Almacenando la distancia, heurística y feromonas en cada una. Por otro lado, en cada par de nodos hay dos aristas, una para cada sentido. Pero son tratadas igual, es decir, la cantidad de heurística y feromonas será la misma en ambas.

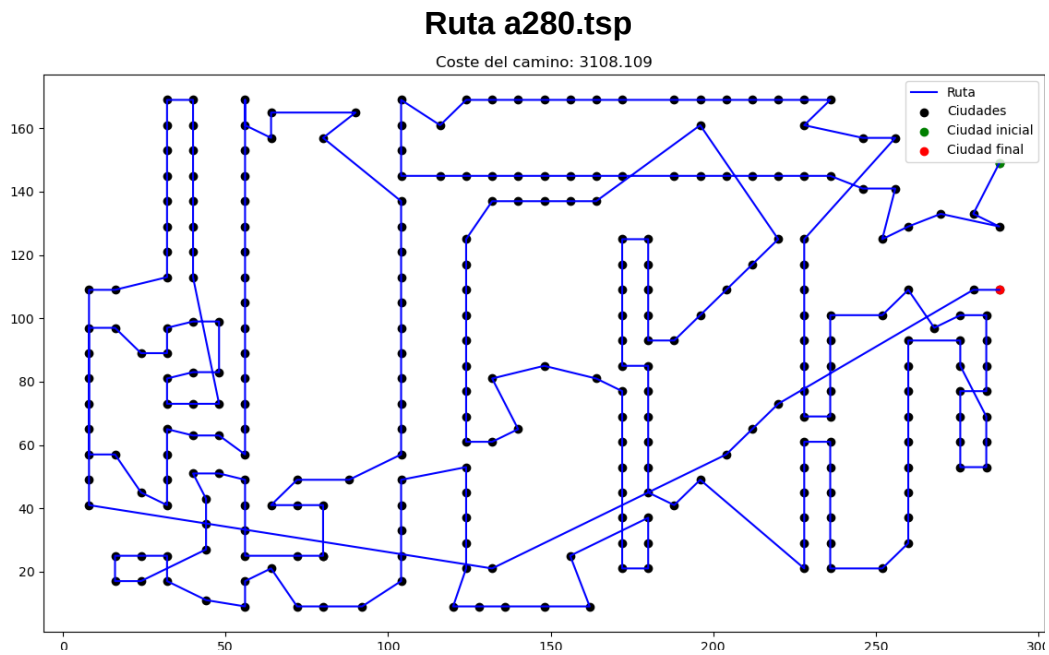
Clase Hormiga

Para recorrer el grafo, se ha desarrollado una clase hormiga. La misma almacenará la ruta seguida hasta el momento, la lista de nodos que aún falta por visitar, la lista de aristas recorridas y el coste de la ruta hasta el momento

Greedy

El algoritmo calculará de forma sencilla una solución inicial. Para ello, tras seleccionar un nodo inicial (el nodo con las primeras coordenadas en el fichero .tsp) se seleccionará el siguiente nodo más cercano. Hasta completar todo el grafo





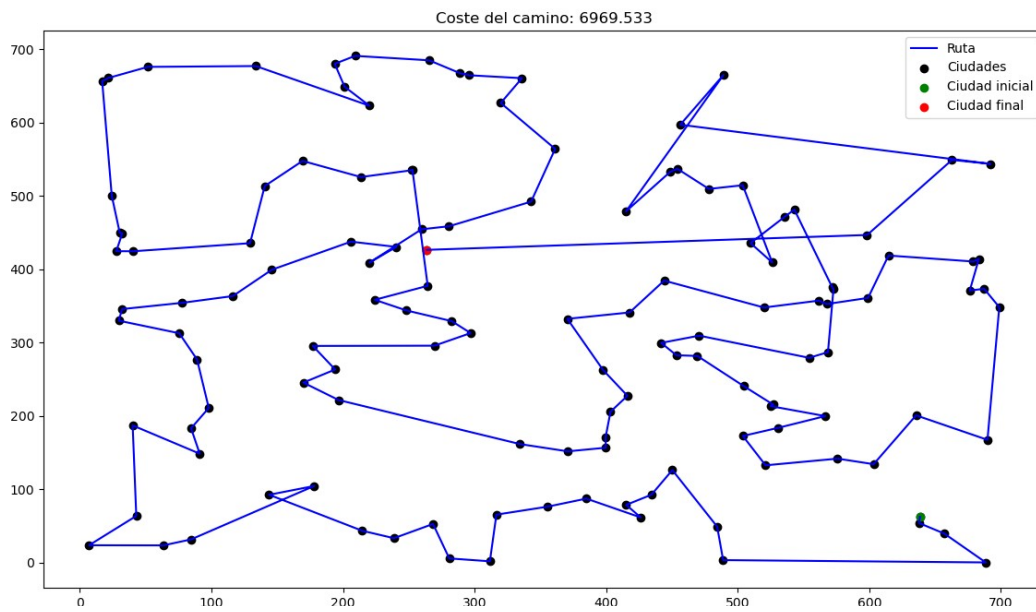
Sistema de Hormigas (SH)

Para la elaboración del código del algoritmo SH, se ha seguido el pseudocódigo mostrado en la teoría. Sin embargo, se ha modificado el apartado de construcción de soluciones. Con el objetivo de aumentar la velocidad del algoritmo. Para ello se ha implementado un método recursivo llamado **ants_find_route(g, ant_list, alpha, beta)**:

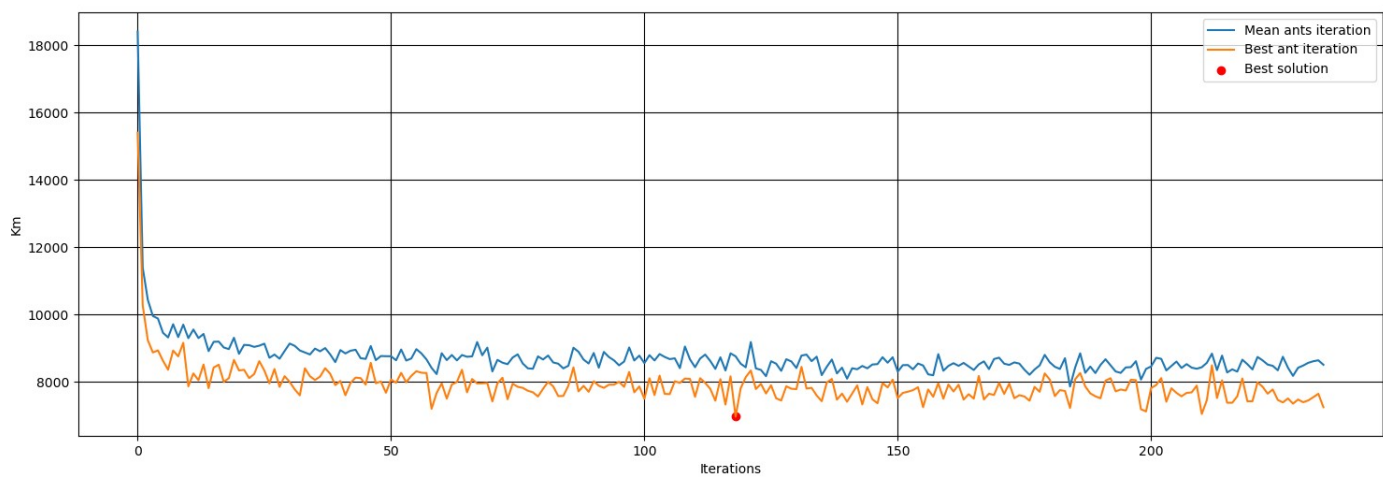
- **g** es el grafo a explorar.
- **ant_list** es la lista de hormigas que recorrerán el grafo.
- **Alpha** es el valor que establece la importancia de la información memorística en la regla de probabilística de transición
- **beta** es el valor que establece la importancia de la información heurística en la regla probabilística de transición

Cada vez que la función sea llamada, cada una de las hormigas buscará el siguiente nodo a seguir. Una vez hecho esto, la función se volverá a invocar. Este proceso se repetirá hasta que las hormigas hayan recorrido el grafo por completo, donde finalmente devolverán el fitness calculado por cada hormiga. Además, en cada una de las hormigas también se almacenará el coste y la ruta tomada. Ejemplo:

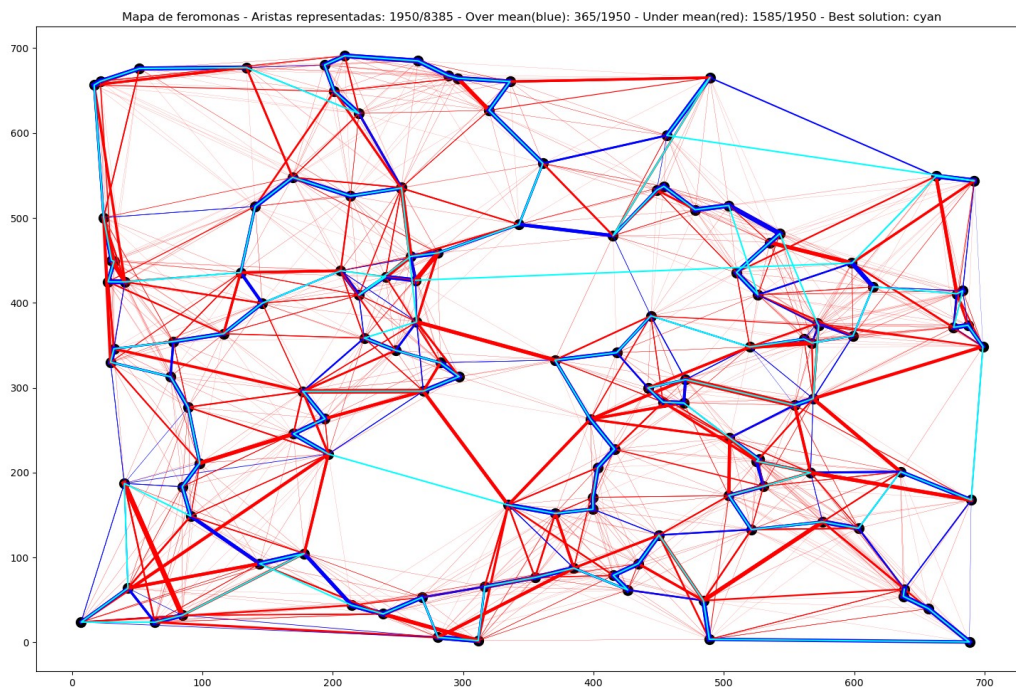
Ruta - ch130.tsp – semilla 55942876



Historial de coste - Ruta - ch130.tsp – semilla 55942876



Mapa de feromonas - ch130.tsp – semilla 55942876

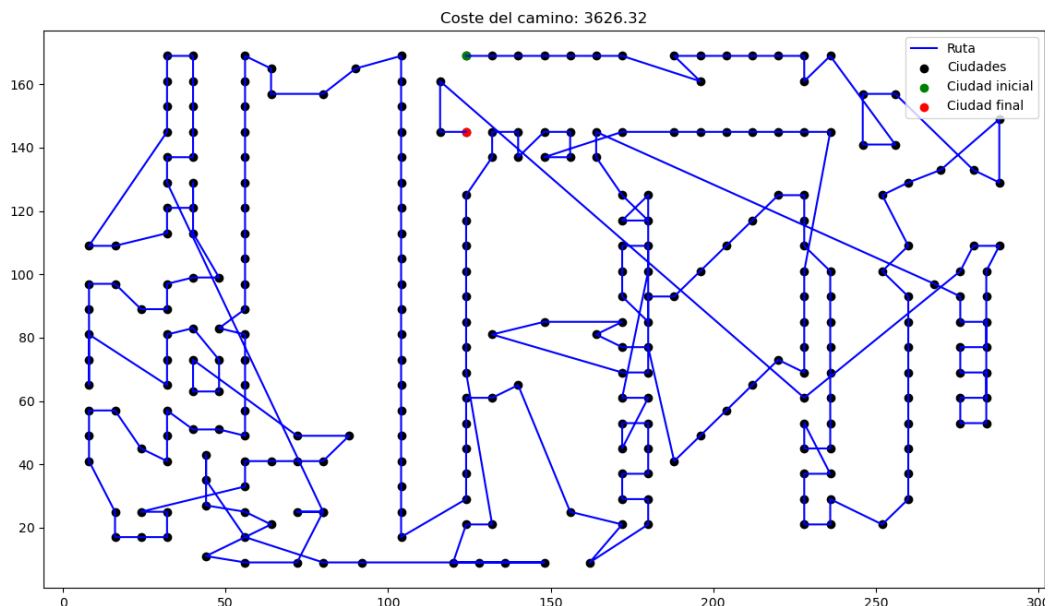


Las líneas rojas y azules representan la cantidad de feromonas que tiene una arista al final de la ejecución del algoritmo. Siendo las de color rojo las que están por debajo de la media(menos prometedoras). Las finas son las que más se acercan al mínimo global y las más gruesas las más cercanas a la media. Por otro lado, las líneas azules representan las más cercanas al máximo de feromonas global(más prometedoras). Las más finas las más cercanas a la media y las más gruesas las más cercanas al máximo global. Cabe aclarar se han representado los 15 nodos más cercanos a cada nodo (con el fin de tener mayor claridad visual). Por último, la línea de color cian representa la ruta tomada por la hormiga.

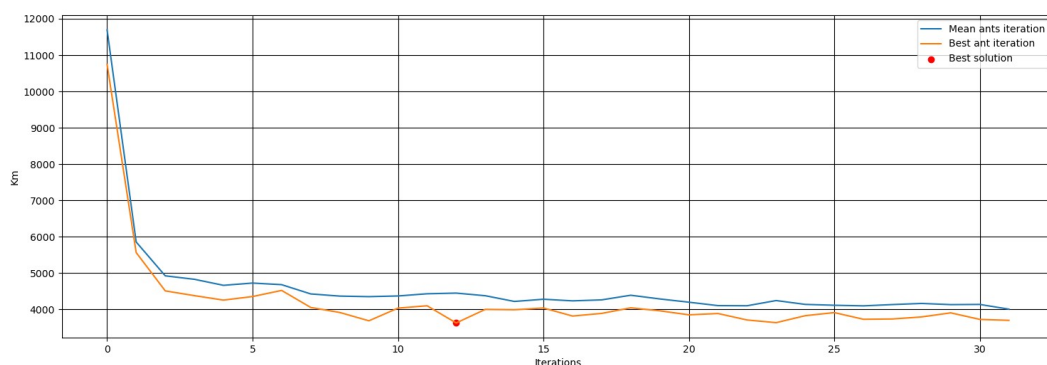
Sistema de Hormigas Elitistas (SHE)

El algoritmo SHE es muy similar al algoritmo SH. Su única diferencia es la regla de actualización que aplica un refuerzo adicional a las aristas que formen parte de la mejor solución encontrada hasta el momento. Ejemplo:

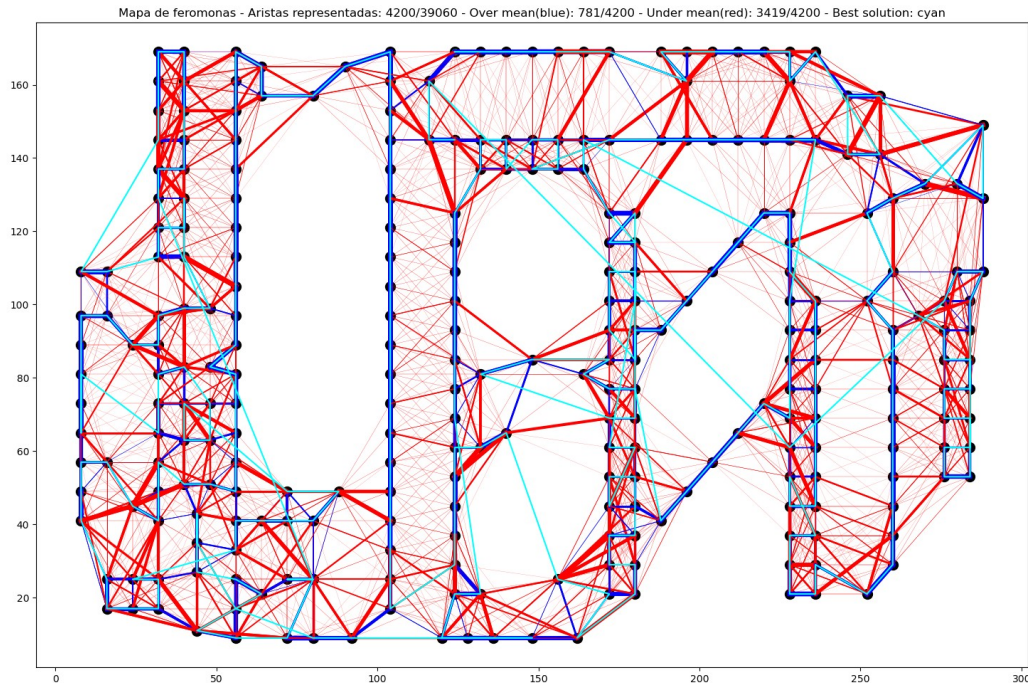
Ruta - a280.tsp – semilla 148263551



Historial de coste - a280.tsp – semilla 148263551



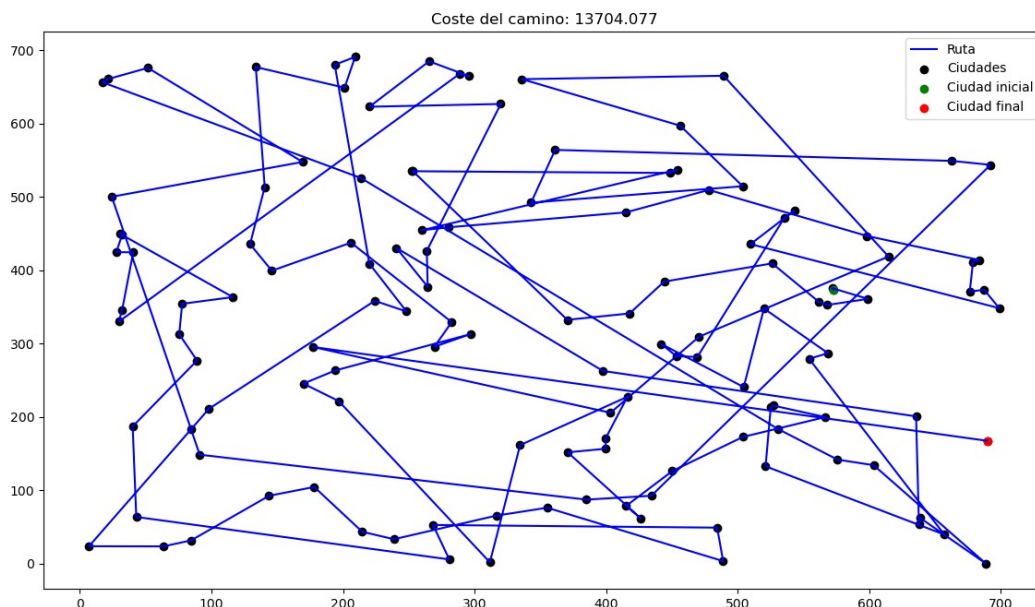
Mapa de feromonas - a280.tsp – semilla 148263551



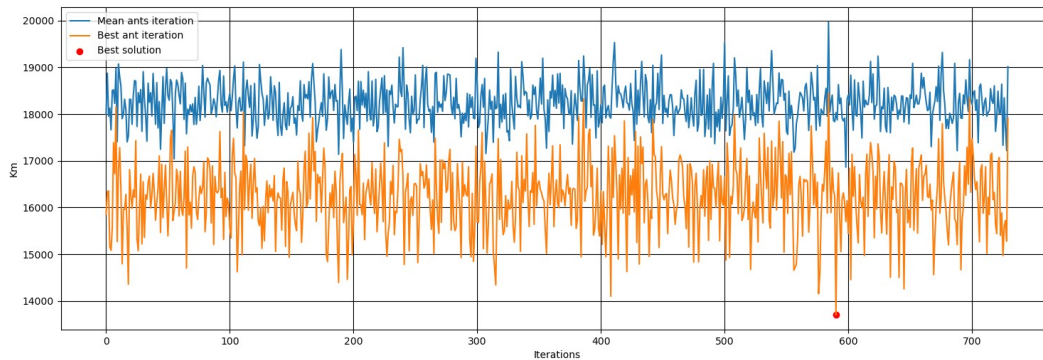
Sistema de Colonias de Hormigas (SCH)

En el algoritmo SCH, se establecerá un equilibrio entre la exploración de nuevas aristas y la explotación de la información acumulada. Habrá dos tipos de generación de feromonas, global y local. Para la primera se seleccionará la hormiga con la mejor solución hasta el momento, y solo se evaporará feromonas en los arcos que componen está. En la segunda, cada hormiga actualizará la feromona de cada arco que visite para diversificar la búsqueda. Ejemplo:

Ruta – ch130.tsp – semilla 148263551



Historial de coste - ch130.tsp – semilla 148263551



Comparación de resultados

Resultados ch130.tsp

Algoritmo	Ev. Medias	Ev. Mejor	Desv. típica Ev	Mejor Fitness	Media Fitness	Desv. típica Fitness
Greedy	1	1	0	7022,12	7022,12	9,09E-13
SH	2353,33	2350	4,71	6777,72	6880,11	78,84
SHE	2340	2330	8,16	6777,72	6880,11	78,84
SCH	7286,67	7260	18,86	13704,08	13794,4	75,39

Resultados a280.tsp

Algoritmo	Ev. Medias	Ev. Mejor	Desv. típica Ev	Mejor Fitness	Media Fitness	Desv. típica Fitness
Greedy	1	1	0	3108,11	3108,11	0
SH	316,67	310	4,71	3389,41	3521,79	98,7
SHE	320	320	0	3389,41	3521,79	98,7
SCH	1830	1800	21,6	8939,9	9244,17	235,84

En el fichero ch130 podemos observar como los algoritmos SH y SHE consiguen mejores resultados que la solución greedy. Pero con un número de evaluaciones muchos mayor. Sin embargo, el algoritmo SH y SHE han tenido el mismo comportamiento. No sabiendo identificar cual es el problema. El algoritmo con peores resultados obtenidos ha sido el SCH, obteniendo resultados considerablemente peores al resto. He presentado problemas al identificar los fallos en este algoritmo(cuando termina una ejecución todas las aristas presentan el mismo valor de feromonas).

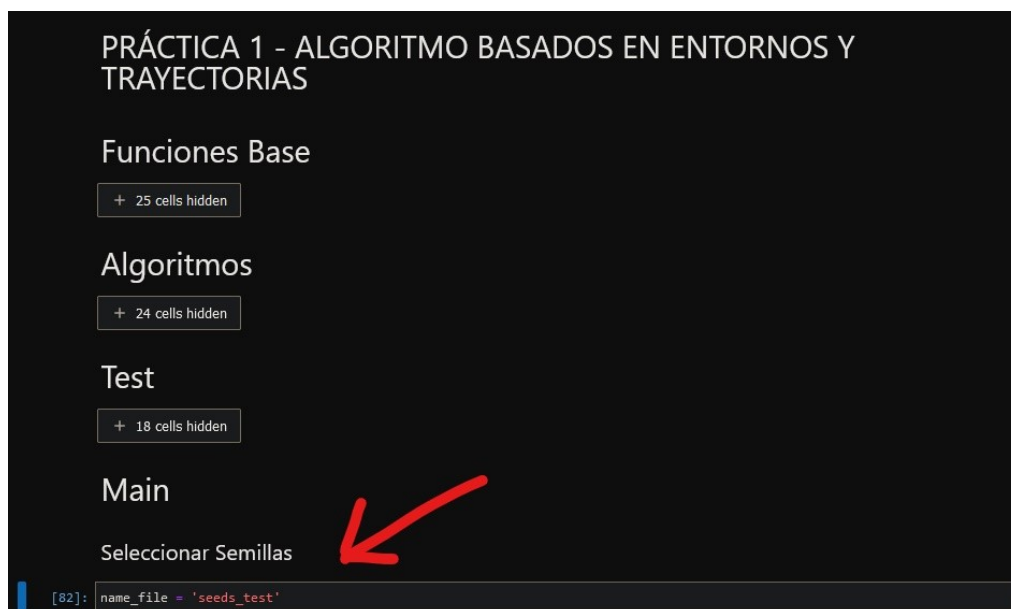
Como ejecutar la práctica

Para ejecutar la practica abriremos el archivo Practica3.ipynb con jupyterLab o Jupyter Notebook.

Una vez hecho eso, el código esta estructurado de forma que a partir de un archivo de semillas en un archivo .csv, se ejecutará la práctica. Este archivo se deberá guardar en la carpeta 'seeds'. Un archivo seed tiene la siguiente estructura:

Orden	Semilla
0	876643
1	486432
...	...
n	435468

Escribiremos el nombre del archivo .csv sin la extensión en la siguiente variable:



Una vez hecho esto, podremos ejecutar el notebook entero.

Para su correcto funcionamiento será necesario instalar la librería networkx:

```
pip install networkx
```

Esta clase nos permitirá crear los grafos de la práctica.