

# Práctica 3 “Algoritmos Voraces (Greedy)”

Parte II: El problema del viajante de comercio



# Índice

1. Objetivo del problema
2. Explicación algoritmo 1: Vecino más cercano.
3. Explicación algoritmo 2: Inserción.
4. Explicación algoritmo 3: Propuesta del grupo, 2-opt
5. Estudio comparativo de algoritmos.



# 1. Objetivo del problema

El objetivo de esta práctica es implementar 3 programas que proporcione soluciones para el problema del viajante de comercio empleando las heurística del vecino más cercano, inserción, así como otra adicional propuesta por el propio equipo. Además hay que realizar un estudio comparativo de las tres estrategias empleando un conjunto de datos de prueba.



## 2. Explicación algoritmo 1: Vecino más cercano.

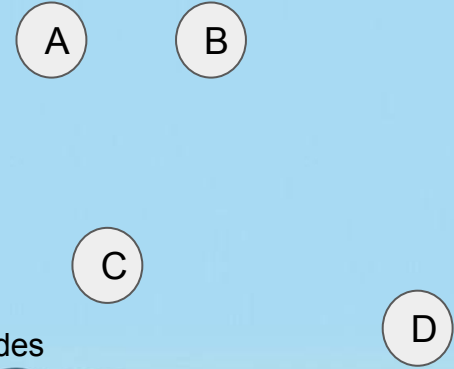
### VECINO MÁS CERCANO

- Conjunto de candidatos (C):** Diferentes ciudades a visitar
- Conjunto de seleccionados (S):** Ciudades ya visitadas
- Función solución:** Esta función tiene que verificar que se han visitado todas las ciudades.
- Función selección:** Aquella función que devuelve la ciudad con menor distancia a la ciudad actual del conjunto de ciudades aún no visitadas.
- Función objetivo:** Minimizar la distancia total del recorrido.



## 2. Explicación algoritmo 1: Vecino más cercano.

- Disponemos de 4 ciudades : A , B , C y D
- Suponemos que empezamos por la ciudad A
- El algoritmo calcula la ciudad más cercana , en este caso B
- Nos encontramos en B ahora y volvemos a calcular la ciudad más cercana
- Realizamos este proceso hasta que recorremos todas las ciudades
- Una vez recorridas todas las ciudades calculamos la distancia total
- Realizamos este mismo procedimiento empezando desde todas las diferentes ciudades
- Comparamos las distancias totales recorridas y nos quedamos con la mejor



# 3. Explicación algoritmo 2: Inserción.

## INSERCIÓN

- Conjunto de candidatos (C):** Diferentes ciudades a visitar
- Conjunto de seleccionados (S):** Ciudades ya visitadas
- Función solución:** Esta función tiene que verificar que se han visitado todas las ciudades.
- Función selección:** Aquella función que devuelve la ciudad, la cual si la añadimos al ciclo de una distancia menor.
- Función objetivo:** Minimizar la distancia total del recorrido.



### 3. Explicación algoritmo 2: Inserción.

- Disponemos de un conjunto de ciudades.
- Partimos inicialmente de un recorrido formado por las ciudades más al norte más al este y más al oeste.
- De entre todas las ciudades no seleccionadas elegimos la que haga que el recorrido de nuestro algoritmo sea menor.
- Formamos todos los distintos caminos posibles con la nueva ciudad seleccionada y nos quedamos con el recorrido el cual la distancia del circuito sea mejor.
- Se repite este proceso hasta que hayamos insertado todas las ciudades en el circuito .



# 4. Explicación algoritmo 3: Propuesta del grupo, 2-opt

-Partimos de la solución obtenida del Vecino más cercano: A B C D

-Tenemos dos contadores  $i, j$ , de manera que  $i$  toma valores de 0 a  $n-1$  y  $j$ , para cada valor de  $i$  toma valores de  $i+1$  hasta  $n$ .  
[ $n$  es el tamaño de la solución]

-Intercambia los valores correspondientes de  $i$  y  $j$  y prueba si dicha solución es un resultado con una distancia total menor.

-Ejemplo:  $i = 0 = A, j = 2 = C \rightarrow$  Resultado = C B A D.

-Comprueba si la solución creada es un resultado con una distancia menor. Si no lo es, vuelve a incrementar los valores de  $i, j$  y vuelve a crear otro recorrido.

-Si no se ha hecho mejora ninguna en varias repeticiones del proceso, el algoritmo acaba.

-Si se ha hecho mejora, el algoritmo se repite con la solución mejorada.





# 5. Estudio comparativo de algoritmos.

En este apartado se va a presentar un estudio comparativo para los resultados obtenidos para cada uno de los algoritmos. La forma de comparación será mediante una tabla en la que hemos recogido los resultados para cada uno de los algoritmo.

Además también mostramos una comparación entre las gráficas obtenidas con gnuplot sobre uno de los archivos, observando así de forma visual, cuál de los algoritmos se ha acercado de forma más cercana al resultado óptimo.



# 5. Estudio comparativo de algoritmos.

Aquí podemos observar una tabla donde se han recogido los datos de recorrido obtenidos para cada uno de los algoritmo.

Fichero	Versión óptima	Vecino más cercano	Inserción	2-opt
a280.tsp	2579	3095.17	3049.28	2788.99
ch130.tsp	6110	7198.74	6555.34	6416.98
kroC100.tsp	20749	23566.4	23401.6	21939.5
eil51.tsp	426	505.77	490.75	449.33
lin105.tsp	14379	16959	16285.1	15072.2
tsp225.tsp	3916	4633.22	4499.19	4182.79
eil101.tsp	629	736.36	687.071	690.36
pcb442.tsp	50778	58953	57970.7	54283.3
berlin52.tsp	7542	8182.19	8286.87	7864.44
rd100.tsp	7910	9427.33	9212.22	8655.46

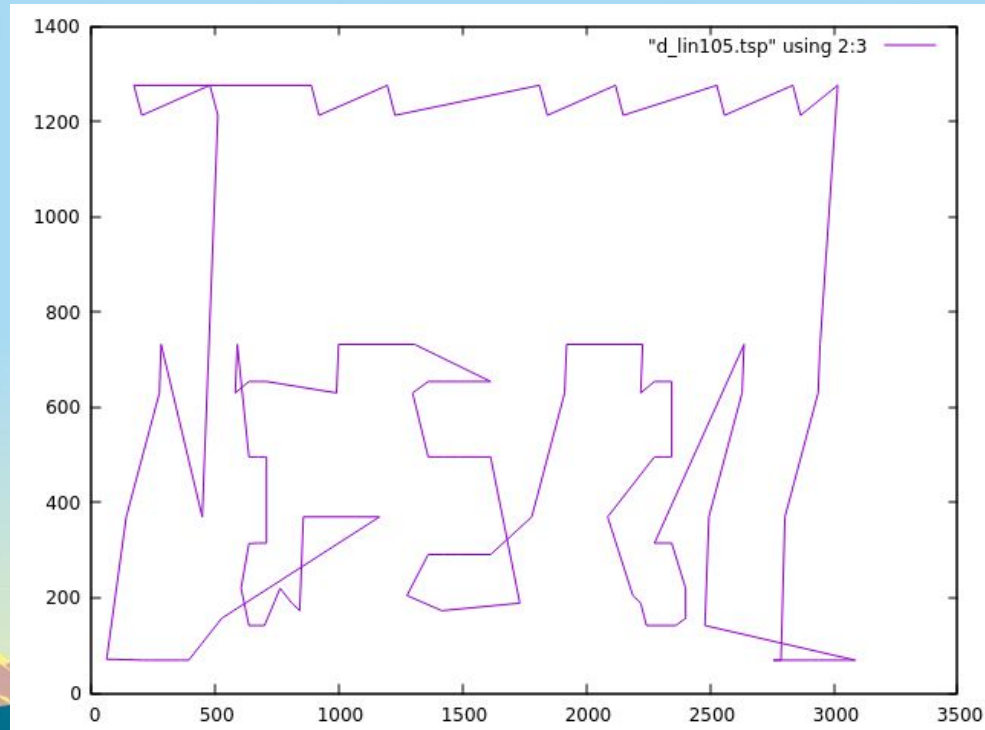
Como observación, destacamos que, los datos obtenidos para el algoritmo 2-opt son mejores y se acercan al valor óptimo



# 5. Estudio comparativo de algoritmos.

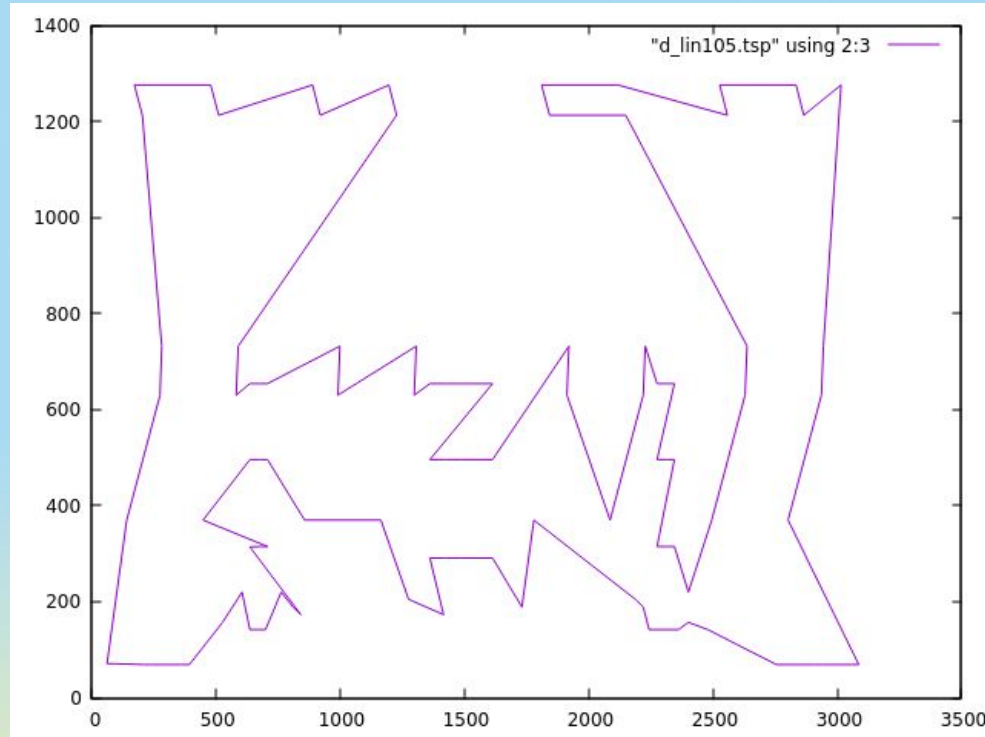
A continuación se muestra, de forma visual, las diferencias entre los datos obtenidos para cada algoritmo. Hemos tomado como ejemplo, la gráfica obtenida para el fichero "lin105.tsp"

## Versión vecino más cercano



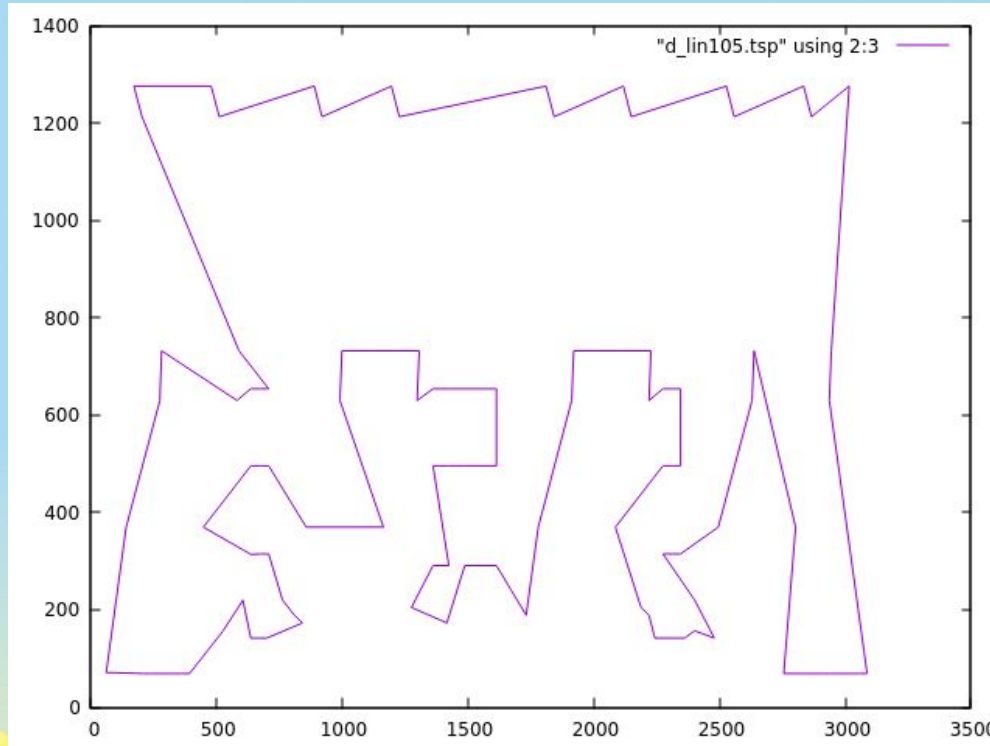
# 5. Estudio comparativo de algoritmos.

## Versión inserción

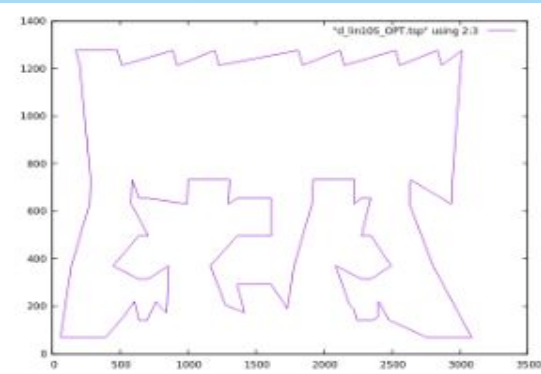


# 5. Estudio comparativo de algoritmos.

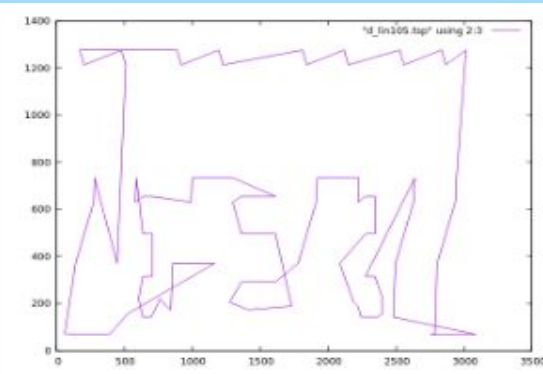
## Versión 2-opt



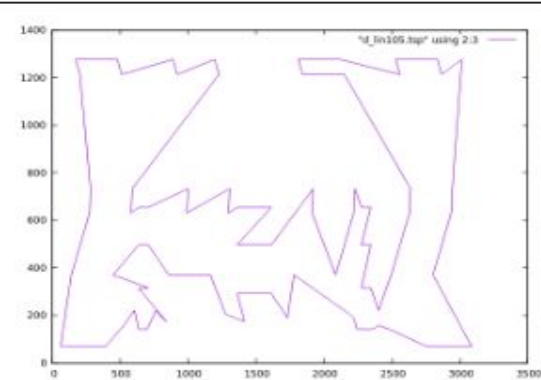
# 5. Estudio comparativo de algoritmos.



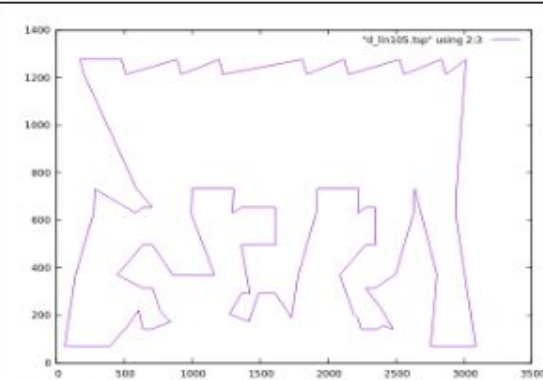
**Versión óptima** (según los datos proporcionados en el .html de la práctica)



**Versión vecino más cercano**



**Versión inserción**



**Versión 2-opt**

Si comparamos las gráficas con la versión más óptima posible (según los datos del .html dado en la práctica), podemos ver que la versión 2-opt es la que más se le acerca.



# Realizado por:

Jesús Baeza Álvarez

Jorge García Moreno

Alejandro Sánchez Molina

David López Maldonado

Jose Manuel Rodríguez Calvo

# FIN

