

Algoritmos Greedy

Pedro Bedmar López

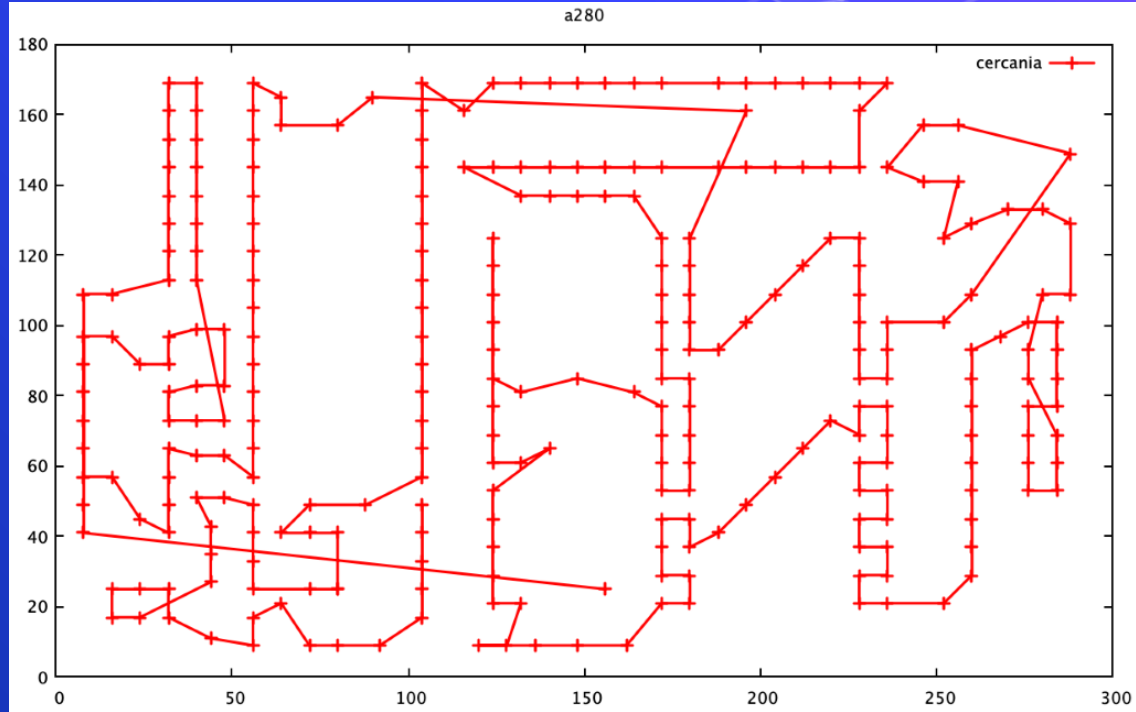
Alejandro Escalona García

Joaquín Alejandro España Sánchez

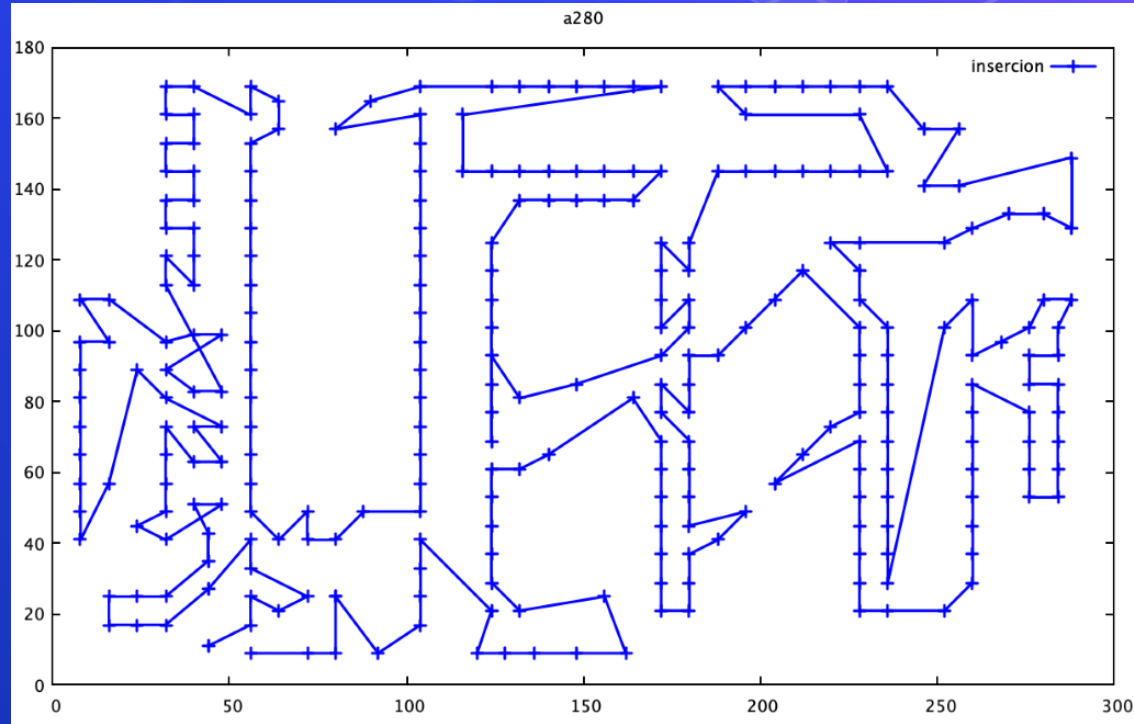
Joaquín García Venegas



Enfoque 1: Cercanía



Enfoque 2: Inserción



Enfoque 3: Propia

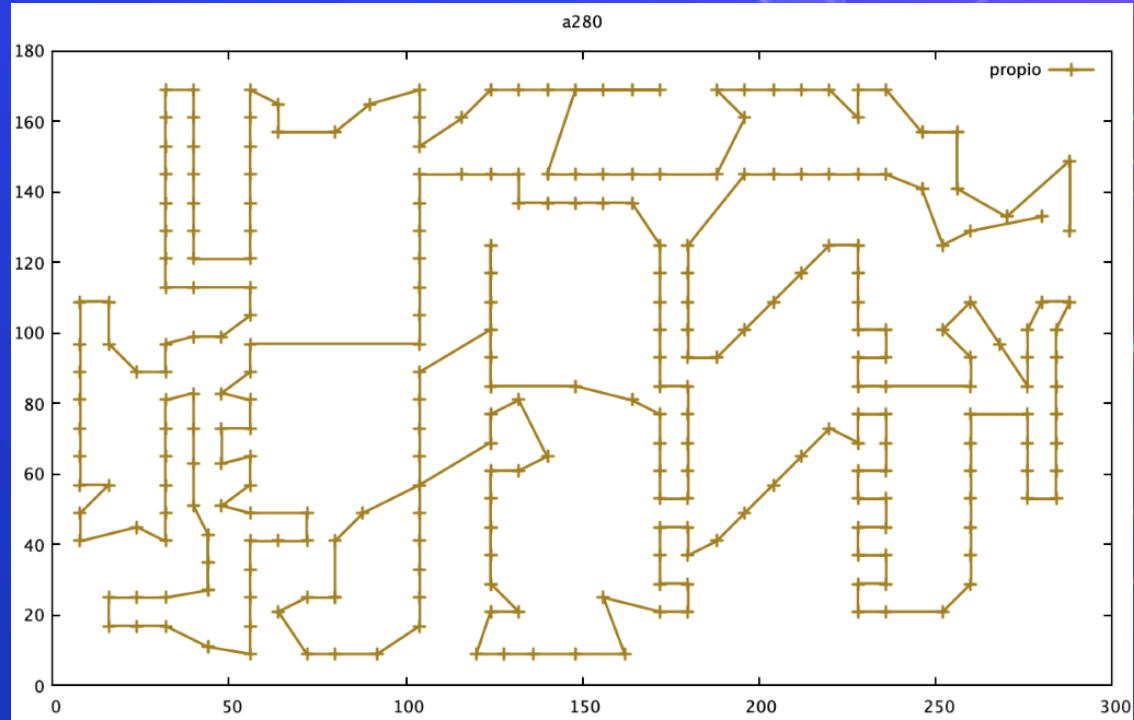


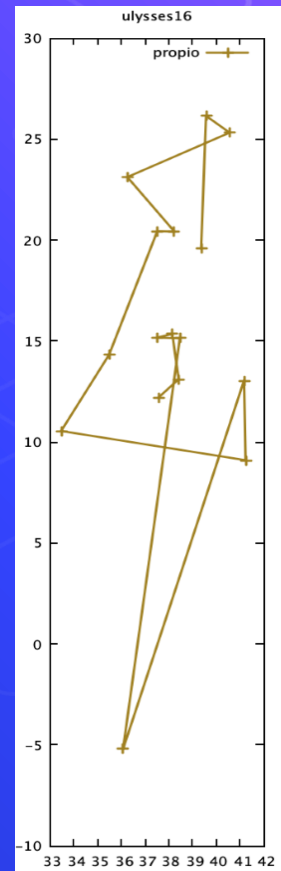
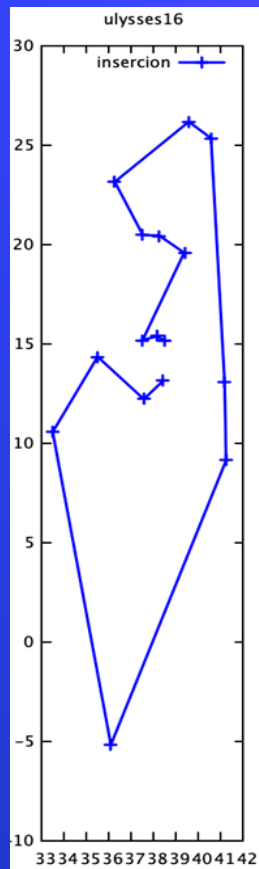
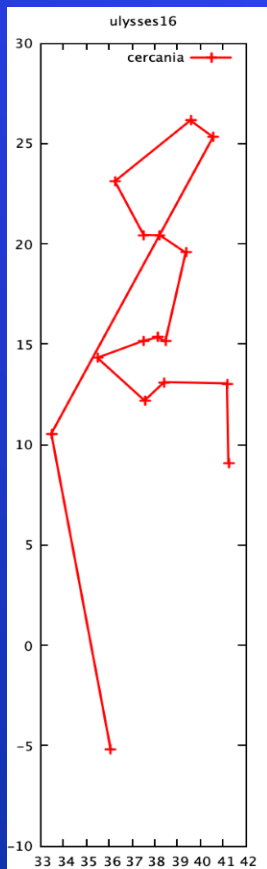
Tabla comparativa de costes

Fichero Tsp/ Coste total del recorrido	Algoritmo Cercanía (media)	Algoritmo Propio(media)	Algoritmo Inserción(media)
a280.tsp	3094.28	2815.32	3051.35
ulysses16.tsp	77.1269	91.51	74.6288
ulysses22.tsp	86.9058	112.82	75.9683
att48.tsp	39236.9	117011	35140.1

Tabla comparativa de tiempos de ejecución

Fichero Tsp/ Tiempo ejecución	Algoritmo Cercanía (media)	Algoritmo Propio(media)	Algoritmo Inserción(media)
a280.tsp	0.552175	18.8016	8.80934
ulysses16.tsp	0.000308	0.007568	0.000756
ulysses22.tsp	0.000747	0.02219	0.002394
att48.tsp	0.008712	0.166956	0.017199

Graficas de los datos de ejecución



Ejercicio 3.1: Planificación de tareas

Vamos a proceder a explicar nuestro algoritmo con un ejemplo:

I	1	2	3	4	5	6
Gi	10	30	20	30	50	20
Di	4	3	1	1	2	2

$\min(6, \max(d_i)) = \min(6, 4) = 4$
Este es el máximo nº de tareas que podrá tener la secuencia.

Ordenamos de forma decreciente de beneficio

I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

⬡ Vamos a proceder a ir introduciéndolas en el conjunto resultante:

I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

Primer intento

	5		
--	---	--	--

Conjunto resultante



I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

Segundo intento

4	5		
---	---	--	--

Conjunto resultante



I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

Tercer intento

4	5	2	
---	---	---	--

Conjunto resultante



I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

Cuarto intento

4	5	2	
---	---	---	--

Conjunto resultante



Como no hay posición libre anterior, se descarta la tarea

I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

Quinto intento

4	5	2	
---	---	---	--

Conjunto resultante

Como no hay posición libre anterior, se descarta la tarea

I	5	4	2	3	6	1
Gi	50	30	30	20	20	10
Di	2	1	3	1	2	4

Sexto intento

4	5	2	1
---	---	---	---

Conjunto resultante

4	5	2	1
---	---	---	---

30 + 50 + 30 + 10 = 120 → Beneficio máximo

Como se puede comprobar, coincide con el resultado de nuestro algoritmo:

```
joaquin@joaquin-X550VX:~/Escritorio/Dropbox/2º-Informatica/Segundo_cuatrimest
/Practicas/P3/asignacionTareas$ ./asignacionTareas 6_tareas
Las tareas introducidas con las siguientes:
I      1      2      3      4      5      6
Gi     10     30     20     30     50     20
Di     4      3      1      1      2      2

La secuencia de tareas es: 4 5 2 1
Con un beneficio total de: 120
```

```
con un beneficio total de: 120
la secuencia de tareas es: 4 5 2 1
```



¿Preguntas?