



**UNIVERSIDAD  
DE GRANADA**

**TSCAO**

**MÁSTER CIENCIA DE DATOS E INGENIERÍA DE COMPUTADORES**

---

# OPTIMIZACIÓN

TRABAJO FINAL

---

**Autor**

Ignacio Vellido Expósito  
ignaciove@correo.ugr.es



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

CURSO 2020-2021

## 1. Parte 1

### El problema de los árboles frutales

Un agricultor dispone de una superficie de  $640\text{m}^2$  para cultivar naranjos, perales, manzanos y limoneros. La cuestión es cómo debe distribuir la superficie entre los árboles para maximizar los beneficios teniendo la siguiente información:

- Los naranjos necesitan un mínimo de  $16\text{m}^2$ , los perales necesitan  $4\text{m}^2$ , los manzanos  $8\text{m}^2$  y cada limonero necesita  $12\text{m}^2$ .
- El agricultor dispone de 900 horas de trabajo al año. Las demandas de los diferentes árboles son: cada naranjo 30 horas, cada peral 5 horas, cada manzano 10 horas y cada limonero 20 horas.
- Debido a las restricciones, el agua disponible para el riego es de  $200\text{m}^3$  al año. La demanda de agua (por año y árbol) es: naranjo:  $2\text{m}^3$ , peral  $3\text{m}^3$ , manzano:  $1\text{m}^3$  y limonero  $2\text{m}^3$ .
- Los beneficios (por año y árbol) son de 50, 25, 20 y 30 euros para los naranjos, perales, manzanos y limoneros respectivamente.

Se plantea el problema de la siguiente forma:

N = Número de naranjos  
M = Número de manzanos  
L = Número de limoneros  
P = Número de perales

$$\begin{array}{ll}\max & 50N + 25P + 20M + 30L \\ \text{s.t.} & 16N + 4P + 8M + 12L \leq 640 \\ & 30N + 5P + 10M + 20L \leq 900 \\ & 2N + 3P + M + 2L \leq 200 \\ & N, P, M, L \geq 0\end{array}$$

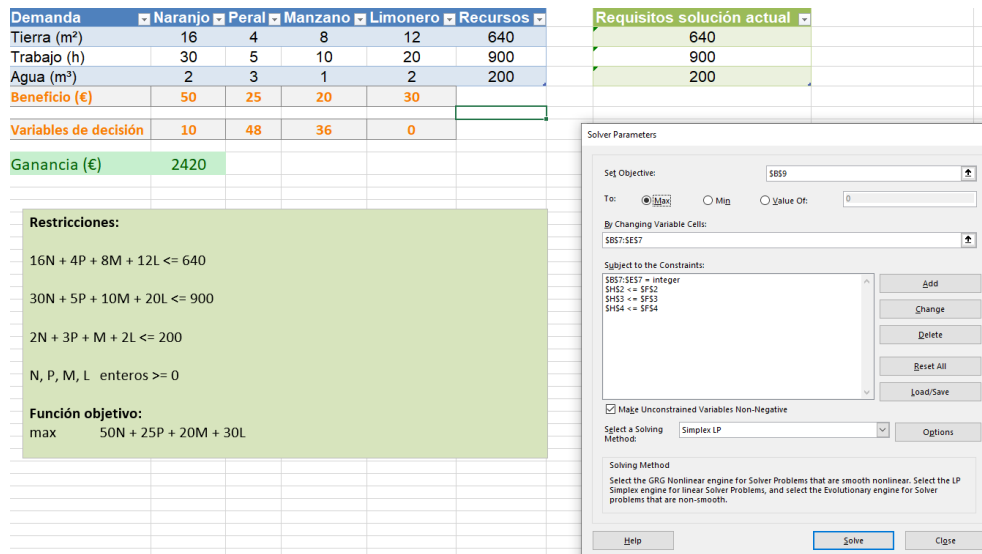


Figura 1: Resolución del problema de los árboles frutales.

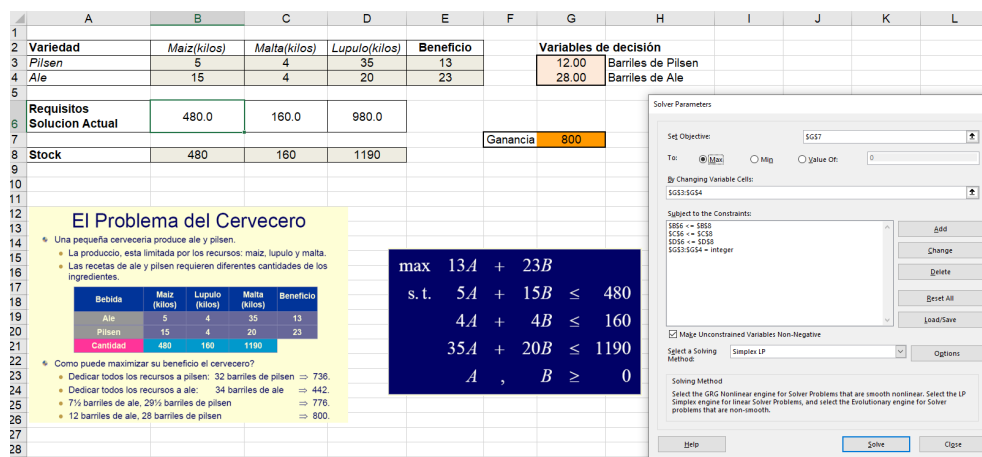


Figura 2: Resolución del problema de la cervecería.

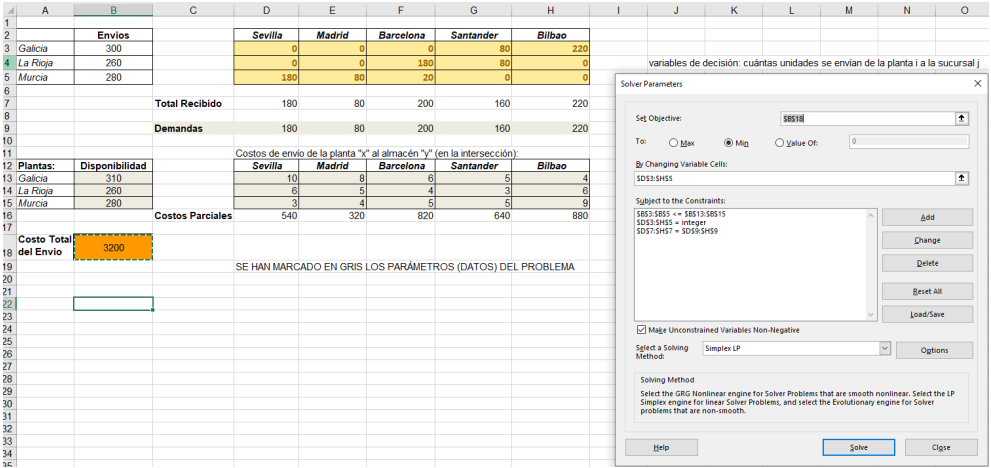


Figura 3: Resolución del problema de la distribución de mercancías.

## 2. Parte 2

| Heurística                       | Problema |         |         |
|----------------------------------|----------|---------|---------|
|                                  | bcl380   | icw1483 | bck2217 |
| Greedy                           | 1862     | 5032    | 7851    |
| Borukva                          | 1915     | 5166    | 7861    |
| Quick Borukva                    | 1872     | 5379    | 8117    |
| Nearest Neighbour                | 2057     | 5681    | 8740    |
| Lin Kernighan Default (+ Greedy) | 1644     | 4487    | 6848    |
| Lin Kernighan 10 kicks           | 1668     | 4738    | 7196    |
| Lin Kernighan 100 kicks          | 1634     | 4601    | 7098    |
| Lin Kernighan + Quick Borukva    | 1630     | 4501    | 6910    |
| Random                           | 26086    | 149247  | 297518  |

Figura 4: Resultados de distancia de cada heurística para los diferentes problemas TSP.

Lin Kernighan obtiene los mejores resultados para cada uno de los problemas, a costa de un mayor tiempo de cómputo. Entre el resto (y a excepción de Nearest Neighbour) no existen diferencias suficientemente significativas en la calidad de las soluciones encontradas, aunque la heurística Greedy es la que obtiene mejores resultados en estas instancias de TSP.

Los cambios en los parámetros de la heurística Lin Kernighan dan resultados inconsistentes, pero apreciamos que el decremento del número de “kicks” (por defecto igual al número de nodos) influye negativamente en la longitud de los caminos encontrados.

También notamos que el cambio del algoritmo inicial afecta en tiempo de cómputo pero apenas en los resultados obtenidos, siendo en general ligeramente inferiores pasando de Greedy a Quick Borukva.

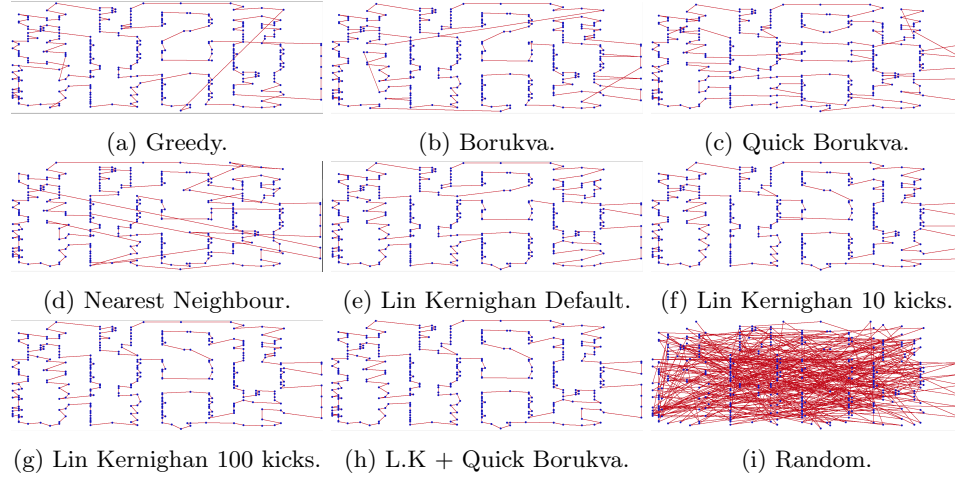


Figura 5: Rutas calculadas por cada heurística para el problema bcl380.

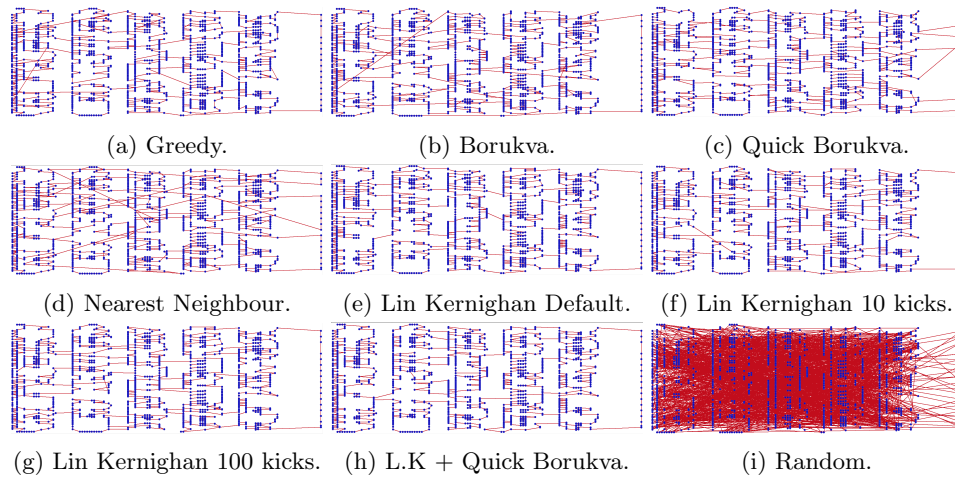


Figura 6: Rutas calculadas por cada heurística para el problema icw1483.

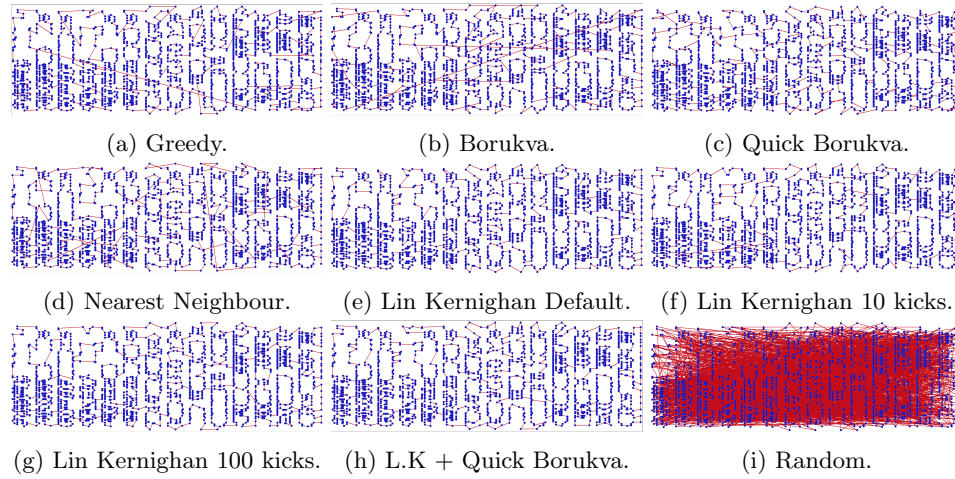


Figura 7: Rutas calculadas por cada heurística para el problema bck2217.