

# RUTA EN CARRETERA



U N I V E R S I D A D  
COMPLUTENSE  
M A D R I D

**ALBERTO VERDEJO**

# Ruta en carretera

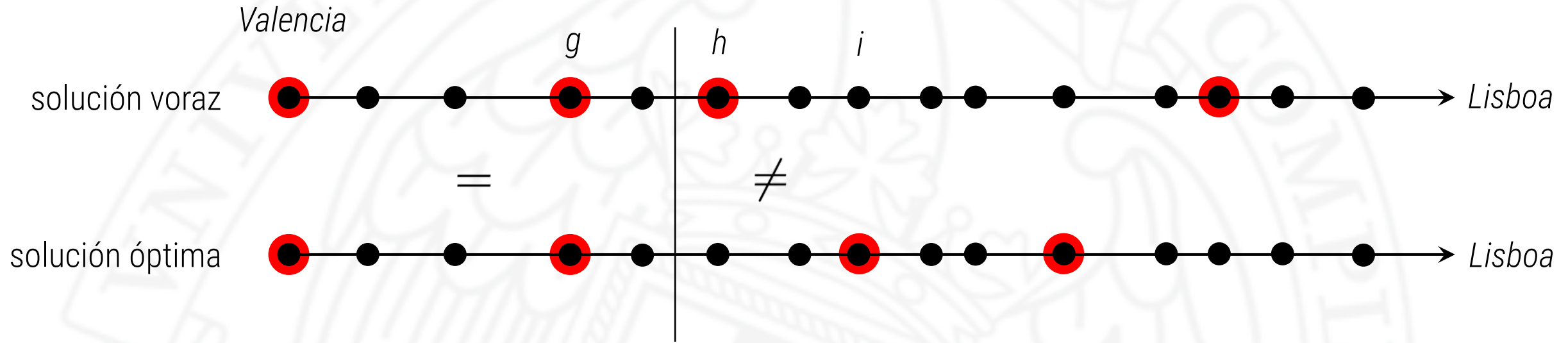
- ▶ Un viajante de comercio tiene que viajar en coche desde Valencia a Lisboa.
- ▶ Con el deposito lleno, su coche puede recorrer un máximo de  $K$  kilómetros.
- ▶ En el mapa de carreteras figuran las distancias entre las gasolineras en su ruta.
- ▶ Parte con el depósito lleno, y quiere realizar un número mínimo de paradas para repostar combustible.



# Ruta en carretera

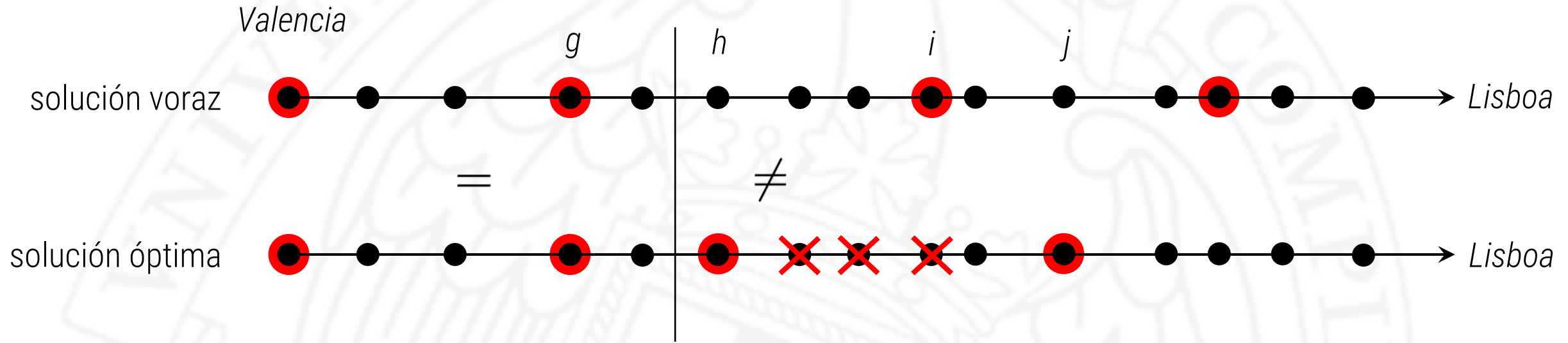
- ▶ Numeramos las gasolineras de 0 a  $n$ , siendo  $g_0$  la de Valencia y  $g_n$  la de Lisboa.
- ▶ Vector  $D[0..n)$  con las distancias entre gasolineras consecutivas de forma que  $D[i] = \text{distancia}(g_i, g_{i+1})$ .
- ▶ Hay solución:  $D[i] \leq K$ , para todo  $i$ .
- ▶ La estrategia voraz consiste en *no parar mientras no haga falta*.

# Ruta en carretera, demostración de la optimalidad



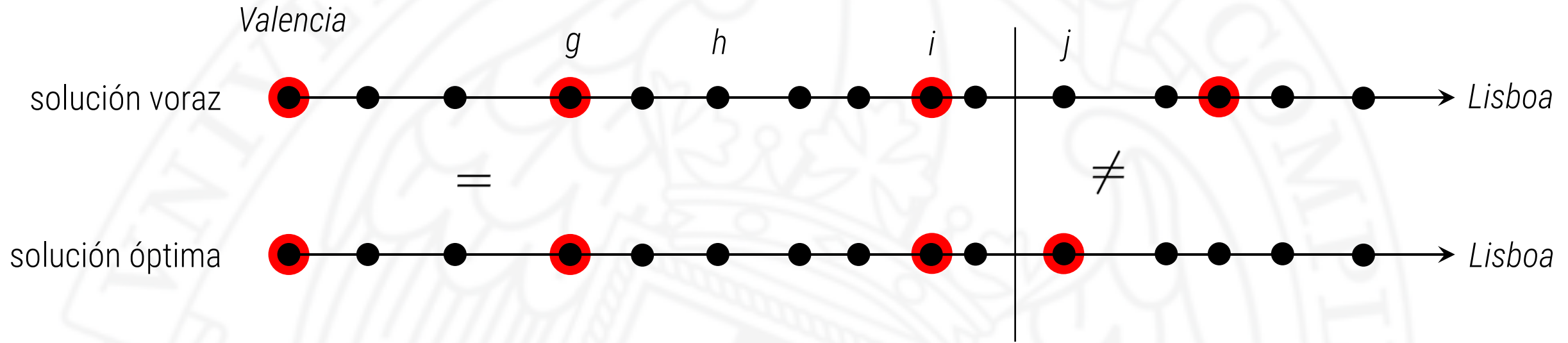
- Esta diferencia **no** puede darse: la solución óptima no cumpliría el test de factibilidad.

# Ruta en carretera, demostración de la optimalidad



- La solución óptima no puede parar en  $\times$

# Ruta en carretera, demostración de la optimalidad



- Retrasamos la parada en la solución óptima.



# Ruta en carretera, implementación

```
int gasolinas(vector<int> const& D, int K, vector<bool> & g) {  
    g[0] = true; // sale con el depósito lleno  
    int paradas = 0;  
    int km = D[0]; // kilómetros andados desde la última parada  
    for (int i = 1; i < D.size(); ++i) { // estamos en  $g_i$   
        if (km + D[i] <= K) {  
            g[i] = false; // no paramos  
        } else {  
            g[i] = true; // hay que parar  
            ++paradas;  
            km = 0;  
        }  
        km += D[i]; // seguimos hasta  $g_{i+1}$   
    }  
    return paradas;  
}
```