RUTA EN CARRETERA

Solución Voraz a este problema.

Tenemos que demostrar la optimalidad de la estrategia voraz.

UNIVERSIDAD COMPLUTENSE MADRID

ALBERTO VERDEJO

Ruta en carretera

- Un viajante de comercio tiene que viajar en coche desde Valencia a Lisboa.

Siguiendo una ruta preestabilizada

- 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.

Distancias entre las gasolineras en SU RUTA.

- Parte con el depósito lleno, y quiere realizar un número mínimo de paradas para repostar combustible.

¿En qué gasolineras debe parar?

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 $D[i]$ = distancia(g_i, g_{i+1}).

D[0] = g0 + g1, distancia de la gasolinera 0 a la 1.

- Hay solución: $D[i] \leq K$, para todo i.

Distancias entre gasolineras adyacentes son menores que los kilometros que puede recorrer con el depósito lleno.

- La estrategia voraz consiste en *no parar mientras no haga falta*.

Candidatos = gasolineras donde podemos parar

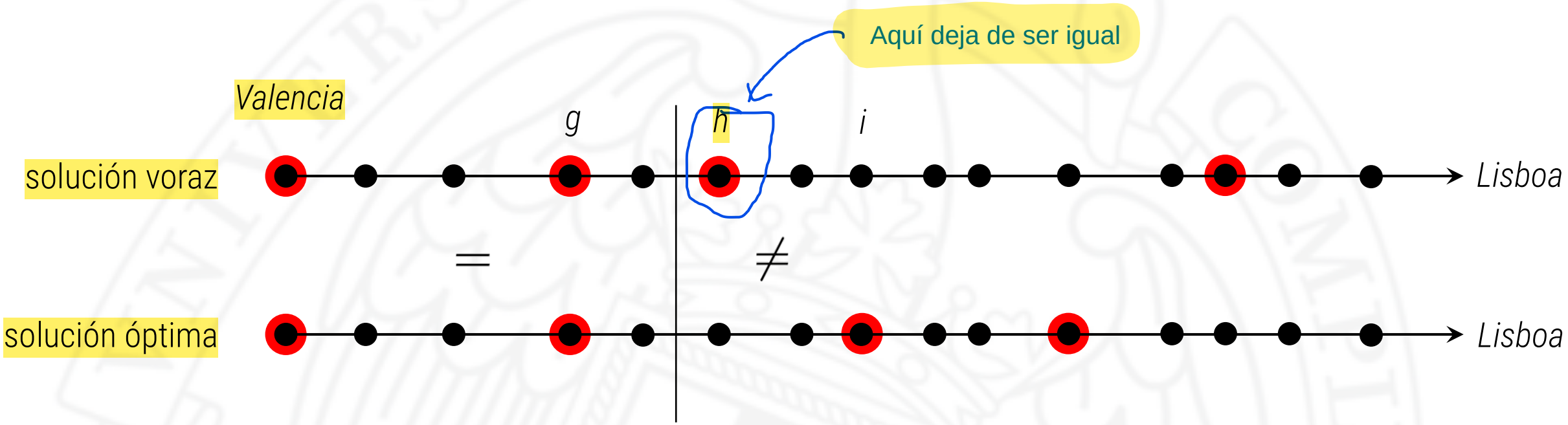
Test de factibilidad= requerir que se pueda recorrer con el coche con el depósito lleno todas las distancias entre paradas consecutivas

Test de solución requiere que hayamos llegado a Lisboa.

Función objetivo = número de paradas. QUEREMOS MINIMIZAR ESTE NÚMERO DE PARADAS.

Al llegar a cada gasolinera, si hay gasolina suficiente en el deposito para llegar a la siguiente se continúa sin parar. y SI NO SE PARA Y SE LLENA EL DEPÓSITO. Ya que queremos minimizar el número de paradas NOL TENDRÍA SENTIDO NO LLENAR EL DEPÓSITO AL COMPLETO CADA VEZ OUE PARAMOS

Ruta en carretera, demostración de la optimalidad



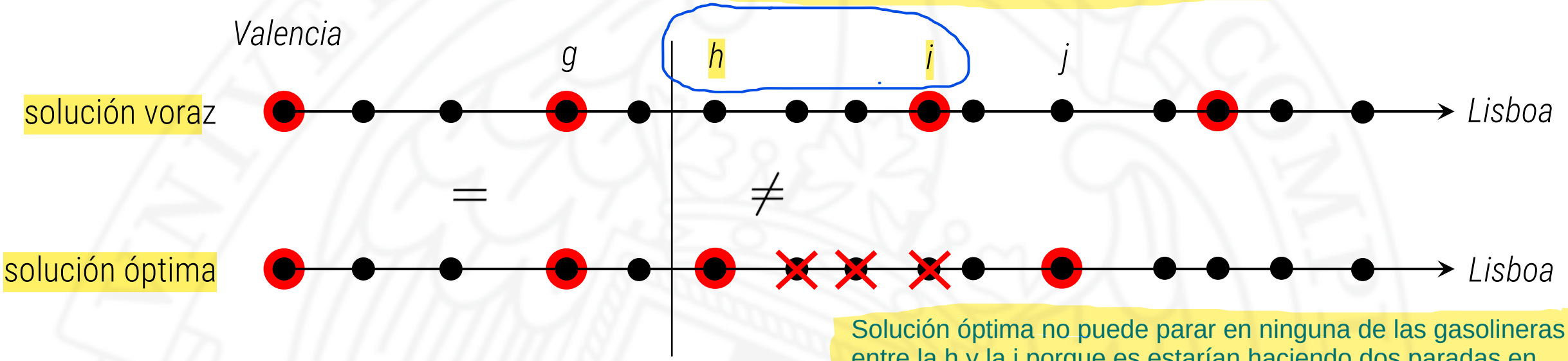
- Esta diferencia no puede darse: la solución óptima no cumpliría el test de factibilidad.

Si el algoritmo voraz ha tenido que parar en la gasolinera h es porque no ha habido más remedio, no podía llegar a la siguiente.

Círculos rojos muestran las paradas que han hecho las dos soluciones.
Círculos negros son las gasolineras.

Ruta en carretera, demostración de la optimalidad

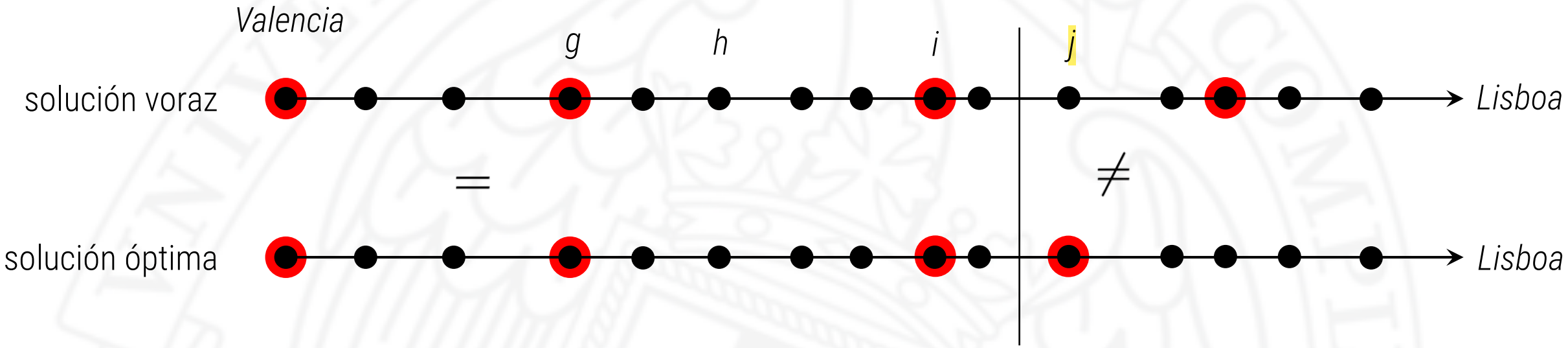
Solución Voraz no puede tener menos paradas que la óptima.



- La solución óptima no puede parar en ✕

Solución óptima no puede parar en ninguna de las gasolineras entre la h y la i porque es estarían haciendo dos paradas en un tramo que se puede recorrer haciendo solamente 1, la última. Desde g a i se puede llegar sin parar, lo comprobó el algoritmo voraz. Por tanto, proponemos retrasar la parada en la gasolinera h hasta la gasolinera i. Haciendo las 2 soluciones iguales hasta esta gasolinera.

Ruta en carretera, demostración de la optimalidad



- Retrasamos la parada en la solución óptima.

Ruta en carretera, implementación

```
int gasolineras(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;
}
```

Vector de distancias entre gasolineras

Kilómetros que podemos recorrer con el depósito lleno

Gasolineras en las que paramos

Devuelve el número de paradas

Se recorren las siguientes gasolineras y en cada una se pregunta si todavía se puede llegar a la siguiente o no.

$O(N)$