

Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

Evaluación Práctica-2: Planificación. 2022-2023.

Nombre: _____

- 1) Subid a Poliformat todos los ficheros .pddl correspondientes a la práctica (habrá distintas versiones: versión original y cada uno de los ejercicios). Se puede subir un archivo .zip.
- 2) Contestad a las preguntas siguientes, rellenando los huecos con las respuestas. Se deben comentar los cambios realizados tanto en el dominio como en el problema. Se debe partir de la práctica ya realizada.
- 3) La entrega fuera de plazo tendrá penalización en la nota.
- 4) La utilización de herramientas de mensajería está terminantemente prohibida.

Tiempo: 75 minutos.

Nota 1: los resultados de TODOS los planes que se obtengan con el planificador LPG deberán realizarse con la misma semilla: -seed 1000

Nota 2: hay que cambiar la función objetivo a la siguiente: (:metric minimize (+ (* 0.3 (total-time)) (* 0.7 (coste-recargas))))

Nota 3: todos los ejercicios y apartados son incrementales. Es decir, el resultado del ejercicio 1 se utiliza como base para el ejercicio 2, el ejercicio 2 se usa como base para el ejercicio 3, y así sucesivamente.

1. (2 puntos, Tiempo estimado: 10') Utilizando vuestro dominio y problema, añadir la siguiente nueva información:

- Un lugar G. La distancia de cualquier lugar A..F a G es de 120 km. La distancia de G a cualquier lugar A..F es de 140 km.
- En G hay un punto de recarga rápida y otro de recarga lenta.
- Hay dos nuevos pasajeros. El pasajero6 debe moverse de A a G. El pasajero7 debe moverse de G a C.

Indica las modificaciones realizadas:

En el problema añadimos la nueva información. En init inicializamos los datos:

(disponible-cargador-rapido G)
(disponible-cargador-lento G)

(at pasajero6 A)
(at pasajero7 G)

(= (distancia A G) 120)
(= (distancia B G) 120)
(= (distancia C G) 120)
(= (distancia D G) 120)
(= (distancia E G) 120)
(= (distancia F G) 120)
(= (distancia G A) 140)
(= (distancia G B) 140)
(= (distancia G C) 140)
(= (distancia G D) 140)
(= (distancia G E) 140)

(= (distancia G F) 140)
(= (distancia G G) 0)

En goal:

(at pasajero6 G)
(at pasajero7 C)

Ejecuta lpg con la opción “-n 3” y muestra el último plan resultante (el que se llama *-pddl_3.SOL):

```
; Version LPG-td-1.0
; Seed 1000
; Command line: lpg-td -o ej1-dominio-taxi.pddl -f ej1-problema-taxi.pddl -n 3 -seed 1000
; Problem ej1-problema-taxi.pddl
; Time 0.42
; Search time 0.42
; Parsing time 0.00
; Mutex time 0.00
; Quality 88.90
```

Time 0.42

```
0.0003: (SUBIR PASAJERO2 TAXI4 B) [1.0000]
0.0005: (CARGAR-LENTO TAXI1 A) [8.0000]
0.0008: (REVISAR TAXI1 A) [2.0000]
1.0010: (MOVER TAXI4 B E) [6.0000]
2.0013: (REVISAR TAXI2 A) [2.0000]
7.0015: (BAJAR PASAJERO2 TAXI4 E) [1.0000]
0.0018: (CARGAR-RAPIDO TAXI2 A) [4.0000]
4.0020: (CARGAR-RAPIDO TAXI2 A) [4.0000]
8.0022: (CARGAR-RAPIDO TAXI2 A) [4.0000]
12.0025: (SUBIR PASAJERO6 TAXI2 A) [1.0000]
13.0028: (MOVER TAXI2 A G) [30.0000]
8.0030: (SUBIR PASAJERO1 TAXI1 A) [1.0000]
9.0033: (MOVER TAXI1 A D) [5.0000]
14.0035: (MOVER TAXI1 D F) [5.0000]
19.0037: (BAJAR PASAJERO1 TAXI1 F) [1.0000]
20.0040: (SUBIR PASAJERO3 TAXI1 F) [1.0000]
21.0042: (MOVER TAXI1 F D) [5.0000]
26.0045: (BAJAR PASAJERO3 TAXI1 D) [1.0000]
0.0047: (CARGAR-RAPIDO TAXI3 C) [4.0000]
4.0050: (MOVER TAXI3 C A) [3.0000]
8.0052: (CARGAR-LENTO TAXI3 A) [8.0000]
16.0055: (MOVER TAXI3 A C) [3.0000]
19.0058: (CARGAR-RAPIDO TAXI3 C) [4.0000]
23.0060: (MOVER TAXI3 C G) [30.0000]
53.0062: (CARGAR-RAPIDO TAXI3 G) [4.0000]
57.0065: (CARGAR-RAPIDO TAXI3 G) [4.0000]
61.0067: (SUBIR PASAJERO7 TAXI3 G) [1.0000]
62.0070: (MOVER TAXI3 G C) [35.0000]
43.0073: (BAJAR PASAJERO6 TAXI2 G) [1.0000]
97.0075: (BAJAR PASAJERO7 TAXI3 C) [1.0000]
98.0078: (REVISAR TAXI3 C) [2.0000]
98.0080: (CARGAR-RAPIDO TAXI3 C) [4.0000]
102.0082: (MOVER TAXI3 C B) [1.0000]
103.0085: (MOVER TAXI3 B D) [4.0000]
107.0088: (SUBIR PASAJERO4 TAXI3 D) [1.0000]
108.0090: (MOVER TAXI3 D E) [8.0000]
```

0.0093: (CARGAR-LENTO TAXI5 D) [8.0000]
 116.0095: (BAJAR PASAJERO4 TAXI3 E) [1.0000]
 8.0098: (MOVER TAXI5 D A) [5.0000]
 13.0100: (MOVER TAXI5 A B) [2.0000]
 15.0103: (MOVER TAXI5 B D) [4.0000]
 19.0105: (CARGAR-LENTO TAXI5 D) [8.0000]
 27.0107: (MOVER TAXI5 D E) [8.0000]
 35.0110: (SUBIR PASAJERO5 TAXI5 E) [1.0000]
 36.0112: (MOVER TAXI5 E A) [15.0000]
 51.0115: (BAJAR PASAJERO5 TAXI5 A) [1.0000]

2. (3 puntos, Tiempo estimado: 20') A partir de ahora, los pasajeros deben pagar por cada transporte realizado. Cada pasajero dispondrá de un bono de transporte en el que cargar billetes para utilizar los taxis. La empresa conseguirá un beneficio ("beneficio-empresa") que habrá que modelar e incluir en la función objetivo, que pasará a ser:

(:metric minimize (- (+ (* 0.3 (total-time)) (* 0.7 (coste-recargas))) (beneficio-empresa)))

El consumo de los billetes se realiza por adelantado, es decir cuando el pasajero sube al taxi. Evidentemente, un pasajero no podrá desplazarse si no dispone al menos de 1 billete en su bono. El transporte siempre consume 1 billete, independientemente del recorrido. Existen dos formas de recargar billetes:

- Recargar un pack de 5 billetes en el bono. Esta acción se realiza en unas máquinas expendedoras instaladas en los lugares A, B y C, a los que será necesario acudir. La duración de la acción es de 1 unidad y el beneficio que obtiene la empresa es de 4 unidades por recarga.
- Recargar 1 billete individual en el bono. Se trata de una experiencia piloto, en el que la máquina expendedora es cualquiera de los taxis (aunque no sea el que se vaya a utilizar para el transporte). De esta forma, podremos utilizar el taxi más cercano en cada momento como máquina expendedora. Lógicamente, para realizar este tipo de recarga tanto el pasajero como el taxi deben estar en el mismo lugar durante toda la acción. La duración de la acción es de 1 unidad y el beneficio que obtiene la empresa es de 1 unidad por recarga.

En el estado inicial todos los pasajeros parten de un bono con 0 billetes.

Indica las modificaciones realizadas:

En el dominio:

Se crea un nuevo predicado:

(disponible-maquina-expendedora ?l - lugar)

Se crean dos nuevas funciones:

(bono ?p - pasajero)
(beneficio-empresa)

Se definen las dos acciones para recargar billetes:

(:durative-action recargar-1billete
 :parameters (?p - pasajero ?t - taxi ?l - lugar)
 :duration (= ?duration 1)
 :condition (and (at start (disponible ?t))
 (over all (at ?p ?l))
 (over all (at ?t ?l)))
 :effect (and (at start (not (disponible ?t))
 (at end (disponible ?t)))

```

        (at end (increase (bono ?p) 1))
        (at end (increase (beneficio-empresa) 1)))
    )

(:durative-action recargar-pack5billetes
:parameters (?p - pasajero ?l - lugar)
:duration (= ?duration 1)
:condition (and (at start (disponible-maquina-expendedora ?l))
                (over all (at ?p ?l)))
:effect (and (at start (not (disponible-maquina-expendedora ?l)))
              (at end (disponible-maquina-expendedora ?l))
              (at end (increase (bono ?p) 5))
              (at end (increase (beneficio-empresa) 4)))
)

```

Y se modifica la condición de la acción de subir:

```

(:durative-action subir
:parameters (?p - pasajero ?t - taxi ?l - lugar)
:duration (= ?duration 1)
:condition (and (over all (at ?t ?l))
                (at start (> (bono ?p) 0))
                (at start (disponible ?t))
                (at start (> (tiempo-a-revision ?t) 0))
                (at start (at ?p ?l)))
:effect (and (at start (not (disponible ?t)))
              (at start (not (at ?p ?l)))
              (at end (decrease (bono ?p) 1)) ;; podría ser también at start
              (at end (in ?p ?t)))
)

```

En el problema:

Inicialización de la información (lugar de las máquinas expendedoras, bonos de pasajeros y beneficio-empresa):

```

(disponible-maquina-expendedora A)
(disponible-maquina-expendedora B)
(disponible-maquina-expendedora C)
(= (bono pasajero1) 0)
(= (bono pasajero2) 0)
(= (bono pasajero3) 0)
(= (bono pasajero4) 0)
(= (bono pasajero5) 0)
(= (bono pasajero6) 0)
(= (bono pasajero7) 0)
(= (beneficio-empresa) 0)

```

Cambiamos la métrica:

```

(:metric minimize (- (+ (* 0.3 (total-time)) (* 0.7 (coste-recargas))) (beneficio-empresa)))

```

Ejecuta lpg con la opción “-n 3” y muestra el último plan resultante (el que se llama *-pddl_3.SOL):

```

; Version LPG-td-1.0
; Seed 1000
; Command line: lpg-td -o ej2-dominio-taxi.pddl -f ej2-problema-taxi.pddl -n 3 -seed 1000
; Problem ej2-problema-taxi.pddl
; Time 0.44
; Search time 0.44

```

; Parsing time 0.00
; Mutex time 0.00
; Quality 106.10

Time 0.44

0.0003: (RECARGAR-PACK5BILLETES PASAJERO2 B) [1.0000]
0.0005: (RECARGAR-PACK5BILLETES PASAJERO6 A) [1.0000]
0.0008: (MOVER TAXI3 C A) [3.0000]
0.0010: (CARGAR-RAPIDO TAXI4 B) [4.0000]
1.0013: (RECARGAR-PACK5BILLETES PASAJERO1 A) [1.0000]
4.0015: (MOVER TAXI4 B E) [6.0000]
0.0018: (CARGAR-LENTO TAXI2 A) [8.0000]
8.0020: (MOVER TAXI2 A F) [20.0000]
8.0022: (CARGAR-LENTO TAXI1 A) [8.0000]
16.0025: (MOVER TAXI1 A C) [3.0000]
19.0028: (MOVER TAXI1 C F) [8.0000]
27.0030: (MOVER TAXI1 F D) [5.0000]
32.0033: (MOVER TAXI1 D A) [5.0000]
28.0035: (RECARGAR-1BILLETE PASAJERO3 TAXI2 F) [1.0000]
37.0037: (CARGAR-RAPIDO TAXI1 A) [4.0000]
16.0040: (CARGAR-LENTO TAXI3 A) [8.0000]
29.0042: (CARGAR-LENTO TAXI2 F) [8.0000]
37.0045: (MOVER TAXI2 F E) [25.0000]
37.0047: (REVISAR TAXI1 A) [2.0000]
39.0050: (REVISAR TAXI3 A) [2.0000]
41.0052: (SUBIR PASAJERO1 TAXI3 A) [1.0000]
42.0055: (MOVER TAXI3 A F) [20.0000]
62.0057: (RECARGAR-1BILLETE PASAJERO5 TAXI2 E) [1.0000]
62.0060: (BAJAR PASAJERO1 TAXI3 F) [1.0000]
63.0062: (SUBIR PASAJERO5 TAXI4 E) [1.0000]
63.0065: (SUBIR PASAJERO3 TAXI3 F) [1.0000]
64.0068: (MOVER TAXI4 E B) [6.0000]
64.0070: (MOVER TAXI3 F D) [5.0000]
69.0072: (BAJAR PASAJERO3 TAXI3 D) [1.0000]
41.0075: (MOVER TAXI1 A B) [2.0000]
43.0078: (SUBIR PASAJERO2 TAXI1 B) [1.0000]
44.0080: (MOVER TAXI1 B E) [6.0000]
50.0083: (BAJAR PASAJERO2 TAXI1 E) [1.0000]
51.0085: (MOVER TAXI1 E B) [6.0000]
57.0088: (MOVER TAXI1 B A) [2.0000]
59.0090: (CARGAR-RAPIDO TAXI1 A) [4.0000]
63.0093: (MOVER TAXI1 A B) [2.0000]
65.0095: (CARGAR-RAPIDO TAXI1 B) [4.0000]
69.0098: (MOVER TAXI1 B D) [4.0000]
73.0100: (MOVER TAXI1 D B) [4.0000]
77.0102: (CARGAR-RAPIDO TAXI1 B) [4.0000]
81.0105: (MOVER TAXI1 B A) [2.0000]
70.0107: (MOVER TAXI4 B A) [2.0000]
70.0110: (RECARGAR-1BILLETE PASAJERO4 TAXI3 D) [1.0000]
72.0113: (BAJAR PASAJERO5 TAXI4 A) [1.0000]
73.0115: (REVISAR TAXI4 A) [2.0000]
83.0117: (CARGAR-LENTO TAXI1 A) [8.0000]
91.0120: (SUBIR PASAJERO6 TAXI1 A) [1.0000]
92.0123: (MOVER TAXI1 A G) [30.0000]
0.0125: (CARGAR-LENTO TAXI5 D) [8.0000]
91.0127: (CARGAR-LENTO TAXI4 A) [8.0000]
99.0130: (MOVER TAXI4 A D) [5.0000]
104.0133: (SUBIR PASAJERO4 TAXI4 D) [1.0000]

105.0135: (MOVER TAXI4 D A) [5.0000]
 110.0137: (MOVER TAXI4 A E) [15.0000]
 125.0140: (BAJAR PASAJERO4 TAXI4 E) [1.0000]
 122.0143: (BAJAR PASAJERO6 TAXI1 G) [1.0000]
 123.0145: (RECARGAR-1BILLETE PASAJERO7 TAXI1 G) [1.0000]
 8.0147: (MOVER TAXI5 D C) [7.0000]
 15.0150: (CARGAR-RAPIDO TAXI5 C) [4.0000]
 19.0152: (MOVER TAXI5 C B) [1.0000]
 20.0155: (MOVER TAXI5 B G) [30.0000]
 50.0158: (CARGAR-RAPIDO TAXI5 G) [4.0000]
 50.0160: (CARGAR-LENTO TAXI5 G) [8.0000]
 124.0163: (SUBIR PASAJERO7 TAXI5 G) [1.0000]
 125.0165: (MOVER TAXI5 G C) [35.0000]
 160.0168: (BAJAR PASAJERO7 TAXI5 C) [1.0000]

3. (3 puntos, Tiempo estimado: 20') Deseamos gestionar la distancia recorrida en kilómetros por cada pasajero ("distancia-recorrida pasajeroX") para poder incluirla, cuando sea necesario, en la función objetivo. En este ejercicio modificaremos la métrica para incluir la distancia del pasajero1 y pasajero2. Es decir, la métrica será:

(:metric minimize (- (+ (distancia-recorrida pasajero1) (+ (distancia-recorrida pasajero2) (+ (* 0.3 (total-time)) (* 0.7 (coste-recargas)))))) (beneficio-empresa)))

Indica las modificaciones realizadas, **considerando además** que en el plan la distancia-recorrida para el pasajero5 debe ser menor que 40:

En el dominio:

Se añade una nueva función:

(distancia-recorrida ?p - pasajero)

Como la información de la distancia recorrida solo aparece en la acción mover, habría que diferenciar una acción de mover-taxi-vacio y otra mover-taxi-con-pasajero. En la segunda acción se actualizaría distancia-recorrida.

```
(:durative-action mover-vacio
:parameters (?t - taxi ?l1 - lugar ?l2 - lugar)
:duration (= ?duration (/ (distancia ?l1 ?l2) 4))
:condition (and (at start (at ?t ?l1))
  (over all (disponible ?t)) ;; aquí el taxi está disponible (luego no tiene pasajero)
  (at start (>= (carga-restante ?t) (distancia ?l1 ?l2))))
:effect (and (at start (not (at ?t ?l1)))
  (at end (at ?t ?l2))
  (at end (decrease (carga-restante ?t) (distancia ?l1 ?l2))))
)
```

```
(:durative-action mover-con-pasajero
:parameters (?t - taxi ?l1 - lugar ?l2 - lugar ?p - pasajero)
:duration (= ?duration (/ (distancia ?l1 ?l2) 4))
:condition (and (at start (at ?t ?l1))
  (over all (in ?p ?t)) ;; aquí el taxi tiene a un pasajero subido
  (at start (>= (carga-restante ?t) (distancia ?l1 ?l2))))
:effect (and (at start (not (at ?t ?l1)))
  (at end (at ?t ?l2))
  (at end (increase (distancia-recorrida ?p) (distancia ?l1 ?l2)))
  (at end (decrease (carga-restante ?t) (distancia ?l1 ?l2))))
)
```

En el problema:

Se inicializa la distancia-recorrida a 0:

```
(= (distancia-recorrida pasajero1) 0)
(= (distancia-recorrida pasajero2) 0)
(= (distancia-recorrida pasajero3) 0)
(= (distancia-recorrida pasajero4) 0)
(= (distancia-recorrida pasajero5) 0)
(= (distancia-recorrida pasajero6) 0)
(= (distancia-recorrida pasajero7) 0)
```

Se modifica el goal:

```
(< (distancia-recorrida pasajero5) 40)
```

Se modifica la métrica:

```
(:metric minimize (- (+ (distancia-recorrida pasajero1) (+ (distancia-recorrida pasajero2) (+ (* 0.3 (total-time)) (* 0.7 (coste-recargas)))))) (beneficio-empresa)))
```

Ejecuta lpg con la opción “-n 3” y muestra el último plan resultante (el que se llama *-pddl_3.SOL):

```
; Version LPG-td-1.0
; Seed 1000
; Command line: lpg-td -o ej3-dominio-taxi.pddl -f ej3-problema-taxi.pddl -n 3 -seed 1000
; Problem ej3-problema-taxi.pddl
; Time 1.05
; Search time 0.94
; Parsing time 0.01
; Mutex time 0.09
; Quality 109.40
```

Time 1.05

```
0.0003: (CARGAR-LENTO TAXI5 D) [8.0000]
0.0005: (RECARGAR-PACK5BILLETES PASAJERO2 B) [1.0000]
8.0007: (MOVER-VACIO TAXI5 D F) [5.0000]
13.0010: (RECARGAR-1BILLETE PASAJERO3 TAXI5 F) [1.0000]
14.0013: (SUBIR PASAJERO3 TAXI5 F) [1.0000]
0.0015: (MOVER-VACIO TAXI3 C A) [3.0000]
0.0018: (CARGAR-LENTO TAXI2 A) [8.0000]
3.0020: (CARGAR-RAPIDO TAXI3 A) [4.0000]
7.0023: (MOVER-VACIO TAXI3 A D) [5.0000]
12.0025: (RECARGAR-1BILLETE PASAJERO4 TAXI3 D) [1.0000]
13.0028: (SUBIR PASAJERO4 TAXI3 D) [1.0000]
14.0030: (MOVER-CON-PASAJERO TAXI3 D E PASAJERO4) [8.0000]
22.0033: (BAJAR PASAJERO4 TAXI3 E) [1.0000]
0.0035: (REVISAR TAXI2 A) [2.0000]
7.0037: (CARGAR-RAPIDO TAXI2 A) [4.0000]
0.0040: (RECARGAR-PACK5BILLETES PASAJERO6 A) [1.0000]
8.0042: (CARGAR-LENTO TAXI2 A) [8.0000]
1.0045: (SUBIR PASAJERO2 TAXI4 B) [1.0000]
```

11.0048: (CARGAR-RAPIDO TAXI1 A) [4.0000]
 2.0050: (MOVER-CON-PASAJERO TAXI4 B E PASAJERO2) [6.0000]
 8.0052: (BAJAR PASAJERO2 TAXI4 E) [1.0000]
 16.0055: (SUBIR PASAJERO6 TAXI2 A) [1.0000]
 17.0058: (MOVER-CON-PASAJERO TAXI2 A G PASAJERO6) [30.0000]
 47.0060: (BAJAR PASAJERO6 TAXI2 G) [1.0000]
 48.0062: (RECARGAR-1BILLETE PASAJERO7 TAXI2 G) [1.0000]
 49.0065: (SUBIR PASAJERO7 TAXI2 G) [1.0000]
 50.0067: (MOVER-CON-PASAJERO TAXI2 G C PASAJERO7) [35.0000]
 85.0070: (BAJAR PASAJERO7 TAXI2 C) [1.0000]
 15.0072: (MOVER-CON-PASAJERO TAXI5 F D PASAJERO3) [5.0000]
 20.0075: (BAJAR PASAJERO3 TAXI5 D) [1.0000]
 21.0077: (CARGAR-LENTO TAXI5 D) [8.0000]
 29.0080: (MOVER-VACIO TAXI5 D E) [8.0000]
 9.0083: (RECARGAR-1BILLETE PASAJERO5 TAXI4 E) [1.0000]
 37.0085: (REVISAR TAXI5 E) [2.0000]
 39.0088: (SUBIR PASAJERO5 TAXI5 E) [1.0000]
 40.0090: (MOVER-CON-PASAJERO TAXI5 E B PASAJERO5) [6.0000]
 46.0093: (MOVER-CON-PASAJERO TAXI5 B A PASAJERO5) [2.0000]
 48.0095: (BAJAR PASAJERO5 TAXI5 A) [1.0000]
 1.0098: (RECARGAR-PACK5BILLETES PASAJERO1 A) [1.0000]
 2.0100: (REVISAR TAXI1 A) [2.0000]
 15.0103: (SUBIR PASAJERO1 TAXI1 A) [1.0000]
 16.0105: (MOVER-CON-PASAJERO TAXI1 A D PASAJERO1) [5.0000]
 21.0107: (MOVER-CON-PASAJERO TAXI1 D F PASAJERO1) [5.0000]
 26.0110: (BAJAR PASAJERO1 TAXI1 F) [1.0000]

¿Qué pasaría si exigimos que la distancia-recorrida por el pasajero5 en el plan deba ser menor que 30?

Lpg no devuelve plan, pero la búsqueda no termina pues se trata de un proceso de búsqueda incompleto y no determinista

4. (1 punto, Tiempo estimado: 10') ¿Qué cambios serían necesarios para que la duración de la acción de mover dependa del tipo de taxi (por ej., hay taxis más rápidos que otros) y cada taxi tenga un ratio de descarga distinto (por ej., un taxi consume más batería que otro)? Explica razonadamente todos los cambios necesarios, aunque **no es necesario** implementarlos en PDDL.

En dominio:

Definir dos nuevas funciones: (velocidad ?t - taxi) (ratio-descarga ?t - taxi)

En todas las acciones de mover, cambiar la duración:

(= ?duration (/ (distancia ?l1 ?l2) (velocidad-taxi ?t)))

En el efecto, cambiar el decremento de la carga-restante: (at end (decrease (carga-restante ?t) (* (ratio-descarga ?t) (distancia ?l1 ?l2))))

En problema:

Inicializar las velocidades y ratios de cada taxi

(= (velocidad taxi1) 4)

(= (velocidad taxi2) 2)...

(= (ratio-descarga taxi1) 1)
(= (ratio-descarga taxi2) 2)...

Nota. Esto habría que hacerlo para todas las acciones de mover, tanto de taxi vacío como taxi con pasajero

5. (1 punto, Tiempo estimado: 10') ¿Qué cambios serían necesarios para tratar de impedir que un taxi se quede sin batería en mitad de un viaje que implica múltiples transportes/tramos? Por ej., ya no basta con que el taxi tenga suficiente autonomía para recorrer un tramo, sino también para posteriormente poder llegar a una estación de recarga. Explica razonadamente todos los cambios necesarios, aunque **no es necesario** implementarlos en PDDL.

En dominio:

Habría que añadir una nueva función que sea “(distancia-hasta-punto-recarga-mas-cercano ?l1 ?l2 – lugar)” con la distancia desde ?l2 a su punto de recarga más cercano. De esta forma, cuando un taxi vaya a moverse desde ?l1 a ?l2 será necesario que tenga suficiente batería para cubrir al menos la distancia a ese punto de recarga más cercano a ?l2. Así garantizamos que no se queda en mitad de un recorrido.

En problema:

Habría que inicializar todas estas distancias, teniendo en cuenta cuál es el punto de recarga más cercano de cada ?l2