

## Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

### Evaluación Práctica-2: Planificación. 2021-22.

Nombre: ELIAS URIOS ALACREU

Al realizar esta prueba de evaluación ACEPTO la "cláusula de veracidad" por la que no recibiré ni daré ayuda en esta prueba y garantizo la autoría del 100% de los resultados.

El incumplimiento por mi parte de los deberes derivados de las buenas prácticas de honestidad académica podrá dar lugar a la adopción de las medidas contenidas en la Normativa de Integridad Académica del alumnado de la UPV.

- 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 (.doc, .rtf, .txt, etc.)  
Se asume que se parte de la práctica ya realizada.

Tiempo: 1 hora.

**Nota 1:** absolutamente todos los resultados 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 (+ (\* 0.2 (total-time)) (\* 0.8 (total-dinero-extra))))

**Nota 3:** todos los ejercicios son acumulativos. Ej: el ejercicio 3 se basa en el 2, que a su vez se basa en el 1.

1. (2.5 puntos, **Tiempo estimado: 10'**) Utilizando vuestro dominio y problema, añadir dos nuevos camiones (inicialmente en D) y dos nuevos pedidos:

Pedido	Origen	Destino
Pedido6	D	C
Pedido7	E	F

Indica los cambios necesarios:

En el fichero problema\_examen.pddl se añade lo siguiente:

(define (problem entrega) (:domain mercancías)

(:objects

[...]

p6 – pedido

p7 – pedido

c3 – camion

c4 - camion

)

(:init

[...]

(at p6 D)

(at p7 D)

(at c3 D)

(at c4 D)

)

(:goal (and

[...]

(at p6 C)

(at p7 F)

))

(:metric minimize (+ (\* 0.2 (total-time)) (\* 0.8 (dinero-extra))))

)

Utilizad LPG con la opción `–speed` y `–quality` para rellenar la siguiente tabla, indicando la duración (*duration*) y la calidad del plan (*plan quality*).

	duration	plan quality
Opción <code>–speed</code>	237 260	50 54.4
Opción <code>–quality</code>	237 260	50 54.4

**Responde a la siguiente pregunta:** ¿Es posible que la opción `–speed` obtenga mejores resultados que la opción `–quality`? Razona la respuesta.

No es posible, dado que la opción `speed` siempre buscará la solución más rápida mientras que la solución `quality` tratará de obtener la mejor de acuerdo a las métricas propuestas por el problema.

2. (3 puntos, **Tiempo estimado: 15'**) La normativa de la empresa ha cambiado. Al entregar un pedido se requiere una acción adicional (además de 'entregar') que es firmar-entrega de duración 2 y que requiere que el pedido no se mueva del punto de entrega. Hasta que no se realice dicha firma no se puede considerar que el pedido está correctamente entregado.

Indica los cambios necesarios y el plan que se obtiene tras ejecutar LPG con la opción `–quality`.

El plan obtenido es:

**Plan computed:**

**Time: (ACTION) [action Duration; action Cost]**

**0.0003: (TRANSPORTAR-FURGO F2 B D) [D:16.0000; C:0.1000]**

**16.0005: (RECOGER F2 D P4) [D:1.0000; C:0.1000]**

0.0008: (RECOGER C3 D P6) [D:1.0000; C:0.1000]

1.0010: (TRANSPORTAR-CAMION C3 D C) [D:7.0000; C:0.1000]

8.0013: (ENTREGAR C3 C P6) [D:2.0000; C:0.1000]

10.0015: (FIRMAR-ENTREGA P6 C) [D:2.0000; C:0.1000]

0.0018: (RECOGER F1 B P2) [D:1.0000; C:0.1000]

1.0020: (TRANSPORTAR-FURGO F1 B A) [D:4.0000; C:0.1000]

5.0023: (RECOGER F1 A P1) [D:1.0000; C:0.1000]

6.0025: (TRANSPORTAR-FURGO F1 A C) [D:6.0000; C:0.1000]

12.0028: (SWAP F1 C3 C P1) [D:3.0000; C:0.1000]

15.0030: (TRANSPORTAR-CAMION C3 C F) [D:7.5000; C:0.1000]

22.5033: (ENTREGAR C3 F P1) [D:2.0000; C:0.1000]

22.5035: (RECOGER C3 F P3) [D:1.0000; C:0.1000]

24.5037: (TRANSPORTAR-CAMION C3 F D) [D:2.5000; C:0.1000]

27.0040: (ENTREGAR C3 D P3) [D:2.0000; C:0.1000]

29.0042: (FIRMAR-ENTREGA P3 D) [D:2.0000; C:0.1000]

17.0045: (TRANSPORTAR-FURGO F2 D E) [D:30.0000; C:0.1000]

47.0047: (ENTREGAR F2 E P4) [D:2.0000; C:0.1000]

49.0050: (FIRMAR-ENTREGA P4 E) [D:2.0000; C:0.1000]

0.0052: (FIRMAR-ENTREGA P7 F) [D:2.0000; C:0.1000]

24.5055: (FIRMAR-ENTREGA P1 F) [D:2.0000; C:0.1000]

47.0057: (RECOGER F2 E P5) [D:1.0000; C:0.1000]

15.0060: (TRANSPORTAR-FURGO F1 C E) [D:55.0000; C:0.1000]

70.0062: (ENTREGAR F1 E P2) [D:2.0000; C:0.1000]

72.0065: (FIRMAR-ENTREGA P2 E) [D:2.0000; C:0.1000]

49.0067: (TRANSPORTAR-FURGO F2 E A) [D:52.0000; C:0.1000]

101.0070: (ENTREGAR F2 A P5) [D:2.0000; C:0.1000]

103.0072: (FIRMAR-ENTREGA P5 A) [D:2.0000; C:0.1000]

**Solution found:**

**Total time: 0.41**

**Search time: 0.11**

**Actions: 29**

**Execution cost: 2.90**

**Duration: 105.000**

**Plan quality: 23.900**

Para llevarlo a cabo se ha introducido un nuevo predicado y la acción firmar-entregar:

(:predicates

[...]

(firmaentrega ?p - pedido ?z - zona)

)

[...]

(:durative-action firmar-entrega

:parameters (?p - pedido ?z - zona)

:duration (= ?duration 2)

:condition (and (over all (at ?p ?z))

)

:effect (and (at end (firmaentrega ?p ?z))

)

)

3. (3 puntos, **Tiempo estimado: 15'**) Nos interesa conocer el número total de transportes (movimientos) que han realizado todos los camiones para poder usar la métrica (:metric minimize (+ (\* 0.2 (total-time)) (\* 0.2 (total-transportes-camiones)))).

Indica los cambios necesarios y el plan que se obtiene tras ejecutar LPG con la opción -quality.

Añadimos una función al dominio y modificamos la acción del transporte del camión para que incremente el valor de la función:

[...]

(:functions

[...]

(total-transportes-camiones)

)

(:durative-action transportar-camion

[...]

:effect (and (at start (not (at ?v ?z1)))

(at end (at ?v ?z2))

(at start (increase (total-transportes-camiones) 1))

(at start (decrease (kilometraje) (distance ?z1 ?z2)))

)

)

Además, hay que modificar el problema, cambiando la métrica por la del enunciado e inicializando la función:

(:init

[...]

(= (total-transportes-camiones) 0)

)

(:metric minimize (+ (\* 0.2 (total-time)) (\* 0.2 (total-transportes-camiones))))

El plan computado es el siguiente:

**Plan computed:**

**Time: (ACTION) [action Duration; action Cost]**

**0.0003: (RECOGER C3 D P4) [D:1.0000; C:0.1000]**

**0.0005: (TRANSPORTAR-FURGO F1 B A) [D:4.0000; C:0.1000]**

**4.0008: (RECOGER F1 A P1) [D:1.0000; C:0.1000]**

**5.0010: (TRANSPORTAR-FURGO F1 A F) [D:100.0000; C:0.1000]**

**0.0012: (TRANSPORTAR-CAMION C4 D F) [D:2.5000; C:0.2000]**

2.5015: (RECOGER C4 F P3) [D:1.0000; C:0.1000]

3.5017: (TRANSPORTAR-CAMION C4 F D) [D:2.5000; C:0.2000]

6.0020: (ENTREGAR C4 D P3) [D:2.0000; C:0.1000]

8.0022: (FIRMAR-ENTREGA P3 D) [D:2.0000; C:0.1000]

6.0025: (RECOGER C4 D P6) [D:1.0000; C:0.1000]

6.0027: (TRANSPORTAR-CAMION C3 D E) [D:7.5000; C:0.2000]

13.5030: (INCREMENTAR) [D:1.0000; C:0.1000]

14.5033: (TRANSPORTAR-CAMION C4 D C) [D:7.0000; C:0.2000]

21.5035: (SWAP C4 C1 C P6) [D:3.0000; C:0.1000]

24.5037: (ENTREGAR C1 C P6) [D:2.0000; C:0.1000]

26.5040: (FIRMAR-ENTREGA P6 C) [D:2.0000; C:0.1000]

0.0043: (RECOGER F2 B P2) [D:1.0000; C:0.1000]

1.0045: (TRANSPORTAR-FURGO F2 B E) [D:20.0000; C:0.1000]

21.0047: (ENTREGAR F2 E P2) [D:2.0000; C:0.1000]

23.0050: (FIRMAR-ENTREGA P2 E) [D:2.0000; C:0.1000]

13.5052: (ENTREGAR C3 E P4) [D:2.0000; C:0.1000]

15.5055: (FIRMAR-ENTREGA P4 E) [D:2.0000; C:0.1000]

21.0058: (RECOGER F2 E P5) [D:1.0000; C:0.1000]

23.0060: (TRANSPORTAR-FURGO F2 E A) [D:52.0000; C:0.1000]

75.0062: (ENTREGAR F2 A P5) [D:2.0000; C:0.1000]

77.0065: (FIRMAR-ENTREGA P5 A) [D:2.0000; C:0.1000]

105.0068: (ENTREGAR F1 F P1) [D:2.0000; C:0.1000]

107.0070: (FIRMAR-ENTREGA P1 F) [D:2.0000; C:0.1000]

0.0072: (FIRMAR-ENTREGA P7 F) [D:2.0000; C:0.1000]

Solution found:

**Total time: 0.41**

**Search time: 0.03**

**Actions: 29**

**Execution cost: 3.30**

**Duration: 109.000**

**Plan quality: 25.100**

4. (1.5 puntos, **Tiempo estimado: 10'**) Explica de forma razonada (**no hace falta modelar ni probar nada**): ¿Cómo harías si quisieras controlar la carga que tiene la batería de las furgonetas, teniendo en cuenta que cada acción de desplazamiento reduce la carga? ¿Qué cambios y extensiones serían necesarias en el dominio y problema? ¿Y si quisiéramos que al final todas las furgonetas llegaran con una carga mínima?

Podríamos hacerlo a través de una función como la siguiente:

```
(:functions
```

```
[...]
```

```
(energia ?c – camion )
```

```
[...]
```

```
)
```

y cada vez que el camion se desplaza simplemente tendríamos que hacer un decrease con el valor deseado, es decir:

```
(:durative-action transportar-furgo
```

```
[...]
```

```
:effect (and (at start (decrease (energia ?c ) 1)
```

```
)
```

Luego a la hora de definir el problema tendríamos que definir la batería de cada furgoneta en el apartado :init, es decir:

```
(:init
```

```
(energia c1 10) ;10 es un valor de ejemplo
```

```
(energia c2 10)
```

```
;Repetir lo mismo para el resto de furgonetas
```

```
[...]
```

```
)
```

Y para que todas las furgonetas lleguen con un mínimo de batería se puede añadir al goal

```
(:goal
```

```
[...]
```

```
(energia c1 1) ;1 es un valor de ejemplo
```

```
(energia c2 1)
```

```
;Repetir lo mismo para el resto
```

```
)
```

