

## Técnicas, Entornos y Aplicaciones de Inteligencia Artificial

### Evaluación Práctica-2: Planificación. 2020-21.

Nombre: \_\_\_\_\_

*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: minimize (total-time)

1. (2 puntos, **Tiempo estimado: 5'**) Utilizando vuestro dominio y problema (**cambiando la métrica como está en la Nota2**), 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 -speed	141	141
Opción -quality	61	61

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

Yo diría que no, ya que con -speed obtenemos el plan más rápido, y con -quality trata de mejorar la métrica. Si -speed logra el plan óptimo, -quality debería también obtener siempre ese. Por lo que -quality debería dar siempre un plan mejor o igual.

2. (3 puntos, **Tiempo estimado: 20'**) El sistema de transporte en nuestra ciudad se ha mejorado:

- Ahora **en ambos sentidos** entre las ciudades E-C y E-D hay metro+tren.
- Existe una **nueva localización F** que, por simplicidad, está a una distancia de 100 unidades de todas las localizaciones existentes. Además, a F **solo** se puede llegar mediante tren desde A.

Los objetivos del problema se mantienen, excepto Juan que desea ir de A a F. Indica los cambios necesarios y el plan que se obtiene tras ejecutar LPG con la opción -quality.

Cambios:

- Añadir ciudad f a *objects*.
- Definir las distancias de cada ciudad a f y de f a cada ciudad en *init*.
- Añadir las conexiones de metro y tren entre las ciudades de la siguiente manera en *init*: e-c, c-e, e-d, d-e.

Plan (normalmente salía plan quality de 120, pero me ha llegado a surgir uno de 61):

Plan computed:

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

0.0003: (COMPRAR-TICKET JUAN) [D:2.0000; C:0.1000]

2.0005: (VIAJAR-TREN JUAN A F) [D:25.0000; C:0.1000]

0.0008: (COMPRAR-TICKET PEDRO) [D:2.0000; C:0.1000]

2.0010: (VIAJAR-TREN PEDRO E B) [D:30.0000; C:0.1000]

0.0012: (RECARGAR-BONO EVA) [D:1.0000; C:0.1000]

1.0015: (VIAJAR-METRO EVA C B) [D:20.0000; C:0.1000]

21.0018: (VIAJAR-METRO EVA B D) [D:40.0000; C:0.1000]

0.0020: (COMPRAR-TICKET ANA) [D:2.0000; C:0.1000]

0.0022: (VIAJAR-BUS ANA D C) [D:40.0000; C:0.1000]

40.0025: (VIAJAR-TREN ANA C A) [D:20.0000; C:0.1000]

0.0027: (COMPRAR-TICKET MARIA) [D:2.0000; C:0.1000]

0.0030: (VIAJAR-METRO MARIA C B) [D:20.0000; C:0.1000]

20.0033: (VIAJAR-TREN MARIA B E) [D:20.0000; C:0.1000]

Solution found:

Total time: 0.41

Search time: 0.05

Actions: 13

Execution cost: 1.30

Duration: 61.000

Plan quality: 61.000

Plan file: plan\_pfile1\_ex2.pddl.SOL Actions: 13

Execution cost: 1.30

Duration: 61.000

Plan quality: 61.000

Plan file: plan\_pfile1\_ex2.pddl.SOL

3. (4 puntos, **Tiempo estimado: 25'**) **Trabajando a partir del ejercicio 2**, se desea ofrecer una nueva alternativa de transporte para viajar con bicicletas en el tren. Para ello es necesario pagar un pequeño suplemento al billete ordinario. **Primero** se debe comprar el billete de tren como en el problema original y **luego** comprar el suplemento para la bicicleta por un precio de 1€ y duración 1. Necesitaremos almacenar la posición de la bicicleta de cada usuario pero, lógicamente, la bicicleta solo se podrá transportar junto con el usuario. Es decir no se pueden transportar bicicletas sueltas.

Probad esta extensión en el problema original, pero **teniendo en cuenta** que ahora Juan y Pedro viajan con bicicleta. Indica los cambios necesarios y el plan que se obtiene tras ejecutar LPG con la opción -quality.

Cambios:

- Añadir predicado para guardar posición de la bicicleta y si el usuario tiene el suplemento.
- Añadir funciones numéricas para saber el precio y la duración de la compra del suplemento.
- Añadir acción de viajar en tren con bicicleta y comprar el suplemento.

Plan:

Plan computed:

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

0.0003: (COMPRAR-TICKET JUAN) [D:2.0000; C:0.1000]

0.0005: (COMPRAR-TICKET PEDRO) [D:2.0000; C:0.1000]

0.0008: (COMPRAR-TICKET ANA) [D:2.0000; C:0.1000]

0.0010: (VIAJAR-BUS ANA D C) [D:40.0000; C:0.1000]

0.0012: (RECARGAR-BONO EVA) [D:1.0000; C:0.1000]

1.0015: (VIAJAR-METRO EVA C B) [D:20.0000; C:0.1000]  
2.0017: (COMPRAR-SUPLEMENTO JUAN) [D:1.0000; C:0.1000]  
2.0020: (COMPRAR-SUPLEMENTO PEDRO) [D:1.0000; C:0.1000]  
3.0022: (VIAJAR-TREN-BICI JUAN A F BICI1) [D:25.0000; C:0.1000]  
3.0025: (VIAJAR-TREN-BICI PEDRO E B BICI2) [D:30.0000; C:0.1000]  
21.0028: (VIAJAR-METRO EVA B D) [D:40.0000; C:0.1000]  
40.0030: (VIAJAR-TREN ANA C A) [D:20.0000; C:0.1000]  
0.0033: (COMPRAR-TICKET MARIA) [D:2.0000; C:0.1000]  
2.0035: (VIAJAR-TREN MARIA C E) [D:50.0000; C:0.1000]

Solution found:

Total time: 0.41

Search time: 0.02

Actions: 14

Execution cost: 1.40

Duration: 61.000

Plan quality: 61.000

Plan file: plan\_pfile1\_ex3.pddl.SOL

4. (1 punto, **Tiempo estimado: 10'**) Explica de forma razonada (**no hace falta modelar ni probar nada**) cómo harías si quisieras controlar el aforo máximo en los viajes en tren, de forma que cuando se llegue al valor máximo no se permita más viajeros. ¿Qué cambios y extensiones serían necesarias en el dominio y problema?

Dominio:

Habría que añadir un objeto tren y una función numérica para saber cuántos viajeros tiene. Otra función para saber el aforo máximo. A partir de ahí se crearían acciones para subir y bajar al tren, y así contar los pasajeros y controlar que no exceda el límite de aforo.

---

Problema:

Añadir dónde está el tren inicialmente y cuantos viajeros tiene. También hay que definir el aforo máximo.