

Técnicas de los Sistemas Inteligentes

Práctica 2: Planificación Clásica



**UNIVERSIDAD
DE GRANADA**

**Realizado por: Daniel Díaz Pareja
Fecha: 05/06/2018
Universidad de Granada**

Índice

1. Ejercicio 1.....	3
2. Ejercicio 2.....	3
3. Ejercicio 3.....	4
4. Ejercicio 4.....	4
4.1 Modificación de board y debark.....	4
4.2 Embarcar y desembarcar múltiple.....	4
4.3 Duración limitada para los viajes de cualquier avión.....	5
4.4 Problemas propuestos.....	5
4.4.1 Problema 1.....	5
4.4.1 Problema 2.....	6
4.4.1 Problema 3.....	6

1. Ejercicio 1

Para resolver el ejercicio 1, se ha incluido un método en la tarea “transport-person” que resuelve el problema de llevar a una persona a un destino si el avión y la persona están, inicialmente, en ciudades distintas.

Simplemente, comprobamos que dichas ciudades son diferentes con el predicado derivado “different”, y si lo son, movemos el avión a la ciudad donde está la persona. A partir de ahí, se embarca, se vuelve a mover el avión a la ciudad destino y se desembarca.

2. Ejercicio 2

Para resolver el ejercicio 2, se ha incluido un método en la tarea “mover-avión” que permite llenar el depósito al avión para volar.

Comprobamos en las precondiciones que el avión no tiene fuel para viajar, en cuyo caso se llama a la acción “refuel”, que recarga el depósito en la ciudad donde está el avión. A continuación, se llama a la acción volar.

Nota: se ha cambiado el predicado derivado “hay-fuel” para que compruebe el consumo de ir de una ciudad a otra (mediante vuelo lento), de manera que pueda recargar si no tiene suficiente fuel.

3. Ejercicio 3

En este ejercicio se han creado predicados para:

- Ver si se puede ir de una ciudad a otra mediante vuelo lento
- Ver si se puede ir de una ciudad a otra mediante vuelo rápido
- Ver si no se supera el límite total de fuel consumido si el avión fuese de una ciudad a otra mediante vuelo rápido.
- Ver si no se supera el límite total de fuel consumido si el avión fuese de una ciudad a otra mediante vuelo lento.

Se ha añadido la función fuel-limit, que no existía.

Se ha modificado la tarea “mover-avion” añadiendo métodos. El primero comprueba si el avión tiene suficiente fuel para realizar un vuelo rápido entre 2 ciudades y no se supera el

límite total de fuel. En tal caso, se realiza un vuelo rápido.

Si no tiene suficiente fuel, se recarga y después se realiza el vuelo rápido (de nuevo, si no se supera el límite total de fuel).

Después, se procede de la misma manera con otros dos métodos similares a los anteriores, pero con el vuelo lento. De esta manera, damos prioridad al vuelo rápido, y si ya no puede hacerse por problemas de límite de fuel, se pasa a realizar vuelos lentos.

4. Ejercicio 4

4.1 Modificación de board y debark.

Para representar que cada avión tiene una capacidad máxima de pasajeros se ha añadido una nueva función “max-passengers ?a”. Se han modificado las primitivas de board y debark para que aumente/disminuya en 1 el valor de otra función “passengers-aboard” cada vez que suba o baje alguien del avión.

4.2 Embarcar y desembarcar múltiple.

Para lograr que se puedan embarcar y desembarcar múltiples pasajeros, necesitamos utilizar una tarea compuesta recursiva. Esta tarea llama a un método cuyas tareas objetivo son embarcar a una persona y después volver a llamarse a si mismo. De esta forma, continuará hasta que ya no queden personas por embarcar, en cuyo caso se llega al caso base y termina.

Se procede de forma similar con desembarcar, pero en este caso se añade como precondition que el destino de la persona sea la ciudad en la que está el avión.

4.3 Duración limitada para los viajes de cualquier avión.

Para lograr este objetivo, se ha añadido una nueva función “max-duration ?a” que nos dice el tiempo máximo para realizar un viaje de un avión. Para que el planificador lo tenga en cuenta, se ha añadido predicados derivados que nos dicen si se supera esta duración al realizar un viaje, tanto para el rápido como para el lento:

(no-supero-duracion-vuelo-rapido ?a - aircraft ?c1 - city ?c2 - city)

(no-supero-duracion-vuelo-lento ?a - aircraft ?c1 - city ?c2 - city)

Estos predicados se añaden a las precondiciones de los métodos de la tarea “mover-avión”, para que este no vuele a destinos cuya duración excede la duración máxima.

4.4 Problemas propuestos.

En los problemas propuestos, se ha puesto que el tiempo de embarque y desembarque sea 0, ya que si el tiempo está en horas no se pueden poner fracciones de la hora, y parecía excesivo poner 1 hora de embarque/desembarque por persona. Las distancias entre ciudades están basadas en la matriz de distancias reales entre aeropuertos españoles.

4.4.1 Problema 1.

Se ha planteado un problema con 10 personas y 9 aviones:

- Los aviones tienen una duración límite por viaje muy alta (100 horas) ya que el objetivo de este problema es comprobar que se consume el menor fuel posible.
- Cada avión está situado en una ciudad (algunas ciudades no tienen avión).
- Las personas están situadas en distintas ciudades (algunas en las mismas, otras no).
- Cada avión tiene como límite de fuel 4 veces la distancia máxima entre su ciudad de inicio y cualquier otra ciudad.
- Cada avión tiene una velocidad diferente, pero todos los aviones tienen una razón de consumo por viaje lento y rápido igual (1 y 2 respectivamente)
- Cada avión tiene una capacidad de depósito diferente.
- Cada avión puede llevar hasta 20 pasajeros.
- Puede verse con más detalle en el fichero problema1.pddl

Puede consultarse el resultado del plan ejecutándolo (no se ha puesto aquí porque es extenso). El resultado en medidas computacionales es el siguiente:

```
Number of actions: 35 (54)
Expansions: 7922
Generated nodes: 16738
Inferences: 0
Time in seconds: 2.031
Real Time: 2.02413
Used Time: 2.031
System Time: 5.0664e-10
```

4.4.1 Problema 2.

Este problema tiene las mismas características que el anterior, pero esta vez cada avión tiene un límite de duración por viaje (algunos 3 horas, otros 4 horas).

Resultados:

```
Number of actions: 38 (57)
Expansions: 1607
Generated nodes: 3305
Inferences: 0
Time in seconds: 0.422
Real Time: 0.421375
Used Time: 0.422
System Time: -8.9407e-11
```

4.4.1 Problema 3.

Este problema tiene las mismas características que el anterior, pero esta vez se han declarado 20 personas y 10 aviones. También se ha aumentado el límite de fuel de cada avión, ya que hay que hacer muchos transportes. La duración máxima por viaje de cada avión se ha mantenido igual que en el problema anterior.

Resultados:

```
Number of actions: 73 (111)
Expansions: 12186
Generated nodes: 23735
Inferences: 0
Time in seconds: 8.422
Real Time: 8.41938
Used Time: 8.422
System Time: 5.0664e-10
```