

# Práctica de Planificación

Grau IA Q1 Curso 2022-2023

Miembros del grupo

Miró López-Feliu, Oriol

Cruz Rodríguez, Unai

Rivera Monsergas, Nico

# Índice

|                         |           |
|-------------------------|-----------|
| <b>Introducción</b>     | <b>3</b>  |
| <b>Dominio</b>          | <b>4</b>  |
| Estado inicial y final  | 4         |
| Variables               | 4         |
| Recurso                 | 4         |
| Persona                 | 4         |
| Suministro              | 5         |
| Base                    | 5         |
| Asentamiento            | 5         |
| Almacén                 | 5         |
| Rover                   | 5         |
| Funciones               | 6         |
| numPersonas             | 6         |
| combustible             | 6         |
| prioridad               | 6         |
| sumaCombustible         | 6         |
| sumaPrioridad           | 6         |
| Predicados              | 6         |
| Operadores              | 7         |
| Mover                   | 7         |
| Coger Suministro        | 8         |
| Coger Persona           | 9         |
| Dejar Suministro        | 10        |
| Dejar Persona           | 10        |
| <b>Juegos de prueba</b> | <b>12</b> |
| Prueba 1                | 12        |
| Introducción            | 12        |
| Entrada FF              | 12        |
| Salida FF               | 14        |
| Resultado               | 16        |
| Prueba 2                | 17        |
| Introducción            | 17        |
| Entrada FF              | 17        |
| Salida FF               | 19        |
| Resultado               | 20        |
| Prueba 3                | 21        |
| Introducción            | 21        |
| Entrada FF              | 21        |
| Salida FF               | 23        |
| Resultado               | 26        |
| Prueba 4                | 27        |

|   |           |
|---|-----------|
| Introducción  | 27        |
| Entrada FF  | 27        |
| Salida FF   | 29        |
| Resultado   | 31        |
| Prueba 5  | 32        |
| Introducción  | 32        |
| Entrada FF  | 32        |
| Salida FF   | 34        |
| Resultado   | 38        |
| Prueba 6  | 38        |
| Introducción  | 38        |
| Entrada FF  | 38        |
| Salida FF   | 41        |
| Resultado   | 44        |
| Prueba 7  | 46        |
| Introducción  | 46        |
| Entrada FF  | 46        |
| Salida FF   | 49        |
| Resultado   | 52        |
| Prueba 8  | 53        |
| Introducción  | 53        |
| Entrada FF  | 53        |
| Salida FF   | 55        |
| Resultado   | 57        |
| Generador de Juegos de Prueba                           | 58        |
| Introducción  | 58        |
| Algoritmo   | 58        |
| Recopilación de datos                                   | 58        |
| Generación de la estructura del problema                | 58        |
| <b>Estudio de la evolución del tiempo de resolución</b> | <b>61</b> |
| Prueba 1  | 61        |
| Prueba 2  | 63        |
| Prueba 3  | 64        |
| <b>Completado de los niveles</b>                        | <b>65</b> |
| Nivel Básico  | 65        |
| Extensión 1   | 66        |
| Extensión 2   | 69        |
| Extensión 3   | 69        |
| <b>Conclusión</b>                                       | <b>71</b> |

# Introducción

En esta práctica se nos pide desarrollar en Fast Forward v2.3 un planificador de transportes de diferentes recursos en marte, usando rovers de transporte como medio. Queremos poder describir e implementar el dominio, conjuntamente con distintos ejemplos de problema a través de lenguajes de descripción (PDDL).

# Dominio

## Estado inicial y final

Para todas las extensiones el estado final es el mismo, uno en el cual se han satisfecho todas las peticiones que se ha podido con los recursos disponibles. Aun así, el problema tiene distintos estados iniciales y restricciones dependiendo de en qué extensión estemos.

Nivel Básico: En el nivel básico, se empieza con los recursos repartidos entre las bases. Los rovers empiezan en bases, y sin ninguna carga. Pueden llevar lo que quieran, y moverse cuanto quieran.

Extensión 1: A partir de esta versión, se introduce una limitación de carga en los rovers: pueden llevar o bien un suministro, o como máximo dos personas. Se inicializa, para cada rover, la función *numPersonas(rover)* a 0 y el predicado *llevaSuministro(rover)* a falso.

Extensión 2: A partir de esta versión, se debe tener en cuenta el combustible de los rovers, como una nueva limitación del problema. Se inicializa por cada rover una función *combustible(rover)* con un valor mayor que 0, y la función *sumaCombustible* a la suma de los combustibles de todos los rovers.

Extensión 3: En esta última extensión, las peticiones tienen una prioridad del 1 al 3. Inicializamos la función *prioridad* por cada petición, y además la función *sumaPrioridad* a 0.

## Variables

### Recurso

Es tipo object y representa un recurso del problema. Tiene dos subtipos, que se detallan a continuación. Cada recurso puede haber sido entregado (usado para servir una petición), o aún no. Si no ha sido entregado, este se puede transportar con tal de satisfacer alguna petición, pero de otra manera, ya no se puede tocar.

### Persona

Un tipo de recurso en Marte es el personal especializado. Esta variable representa una unidad de ese recurso, y las personas están en bases de tipo asentamiento, como se explicará más adelante.

## Suministro

El segundo tipo de recurso en Marte son los suministros. Esta variable representa una unidad de este recurso, y se necesitará transportar este desde almacenes a asentamientos.

## Base

Es tipo object y representa un nodo en la red de marte. Todas las bases están conectadas de alguna manera, es decir, la red de todas las bases con sus conexiones forma un grafo conexo. Existen dos tipos de base en el problema.

## Asentamiento

Este es el tipo de base en el que viven los colonos. Es decir, es de donde parten, y a donde llegan, los recursos del subtipo Persona.

## Almacén

Este es el tipo de base a donde llegan los suministros de la tierra. Es de donde parten los recursos del subtipo Suministro.

## Rover

Es tipo object, y representa el objeto más importante del problema. Es gracias a los desplazamientos de este por la geografía marciana que podremos satisfacer peticiones. Este objeto varía dependiendo de la versión del problema:

Nivel Básico: En esta versión del problema, los rovers pueden transportar tanto personas como suministros, sin limitación de carga ni ningún otro tipo.

Extensión 1: Esta versión introduce limitaciones de carga y tipo de transporte en los rovers. Ahora solo podrán transportar como máximo una carga de suministros o dos personas. Además, ya no podremos llevar mezclados personas y suministros.

Extensión 2: Esta versión implica todas las restricciones de la extensión 1, y además se añade el concepto de combustible. En cada desplazamiento entre dos bases, el rover gasta una unidad de combustible, y no se podrá mover una vez agotado este. Se recarga al principio de cada día, así que nos llegarán todos los rovers con el depósito lleno. Asumiremos que no todos los rovers tienen la misma capacidad máxima de combustible, ya que quizá existen diferentes modelos, o alguno está averiado.

Extensión 3: Esta es la última versión, no se realiza ningún cambio sobre los rovers, se comportan igual que en la extensión 2.

## Funciones

### numPersonas

Esta función, introducida en la extensión 1, cuenta para cada rover la cantidad de personas que lleva. Es útil para controlar si lleva o no personas, para saber si puede cargar suministros, y si lleva menos de dos personas, para poder cargar personas.

### combustible

Esta función ha sido introducida en la extensión 2. Cuenta, para cada rover, la cantidad de combustible restante de este. Sirve para controlar cuando un rover puede moverse o no.

### prioridad

Esta función ha sido introducida en la extensión 3. Cuenta, para cada petición, su prioridad, que es un valor del uno al tres. Sirve para poder representar el concepto de prioridad en las peticiones.

### sumaCombustible

Esta función ha sido introducida en la extensión 2. Cuenta la suma de combustible restante entre todos los rovers. Sirve para poder minimizar de manera cómoda la cantidad de combustible del problema, sin tener que volverlo a calcular.

### sumaPrioridad

Esta función ha sido introducida en la extensión 3. Cuenta la suma de las prioridades de las peticiones que hemos satisfecho. Es útil para maximizar las prioridades satisfechas, de manera cómoda, sin tener que volverlas a calcular.

## Predicados

Para modelar los problemas, hemos ideado los siguientes predicados:

- *petición(Recurso, Base)*: Indica que existe una petición de un recurso a una base. El planificador buscará satisfacer estas peticiones. Nos es útil para saber qué movimientos debemos realizar.
- *posicionRecurso(Recurso, Base)*: Indica en qué base se encuentra actualmente un recurso. Nos es útil para poder identificar dónde se encuentra cada recurso.

- *posicionRover(Rover, Base)*: Indica en qué base se encuentra actualmente un rover. Nos es útil para poder identificar dónde se encuentra cada rover.
- *basesAdyacentes(Base, Base)*: Indica que dos bases son adyacentes, es decir, que tienen conexión una con otra. Nos es útil para saber a qué base nos podemos mover desde la actual, es decir, qué caminos podemos realizar.
- *transportaSuministro(Rover, Suministro)*: Nos indica que un rover está transportando un suministro específico actualmente. Nos es útil para saber qué suministros está transportando el rover, en caso de que haya alguno.
- *transportaPersona(Rover, Persona)*: De manera análoga al predicado *transportaSuministro*, Nos indica que un rover está transportando una persona específica actualmente. Nos es útil para saber qué personas está transportando el rover, en caso de que haya alguna.
- *llevaSuministro(Rover)*: Introducido en la extensión 1. Nos indica que un rover está llevando actualmente un suministro cualquiera. Es útil para evitar usar un *forall* cuando comprobamos la carga de un rover, ya que de otra manera necesitamos comprobar, por todos los suministros, que no exista ninguno que cumpla *transportaSuministro* con el rover actual.
- *entregado(Recurso)*: Nos indica que uno de los recursos disponibles ha sido entregado, es decir, se ha satisfecho con él una petición. Nos es útil para saber qué recursos hemos utilizado ya, y cuando hemos cumplido la planificación. No podemos considerar la planificación completa una vez satisfechas todas las peticiones, debido a que algunas no se podrán satisfacer al haber más peticiones que recursos, y por tanto, hemos de considerar como hecha la planificación una vez asignados todos los recursos disponibles.

## Operadores

### Mover

Este operador se usa para mover un rover de una base a otra adyacente. Recibe como parámetros el rover que queremos mover y dos bases, la de origen *b1* y la de destino *b2*. Es un operador fundamental, y su utilidad es permitir hacer los movimientos de recursos.

Recibe como parámetros un rover y dos bases. La precondition es que el rover esté en la primera base y ambas bases sean adyacentes. A partir de la extensión 2, se requiere también que el rover tenga una cantidad positiva de combustible, con tal de ser capaz de realizar el movimiento.



El efecto de este operador es que el rover deja de estar en la primera base y pasa a estar en la segunda, y además, a partir de la extensión 2, reducimos en una unidad el combustible del rover.

|              |   |       |
|--------------|---|-------|
| Parámetros   | Rover <i>rover</i> , Base <i>b1</i> , Base <i>b2</i>  |       |
| Precondición | $basesAdyacentes(b1, b2) \wedge posicionRover(rover, b1)$   | Base  |
|              | $basesAdyacentes(b1, b2) \wedge posicionRover(rover, b1) \wedge combustible(rover) > 0$   | Ext 2 |
| Efecto       | $\neg posicionRover(rover, 1) \wedge posicionRover(rover, b2)$  | Base  |
|              | $\neg posicionRover(rover, 1) \wedge posicionRover(rover, b2) \wedge decrease(combustible(rover), 1) \wedge decrease(sumaCombustible, 1)$ | Ext 2 |

Tabla 1: Descripción del operador *mover*

## Coger Suministro

Este operador se encarga de cargar un suministro de una base en un rover. Recibe como parámetros un rover, una base y un suministro, y tiene como precondición que tanto el rover como el suministro se encuentren en la base indicada, y el suministro aún no haya sido entregado a nadie. Además, a partir de la extensión 1, el rover no puede llevar antes de coger el suministro ningún otro suministro ni ninguna persona.

El efecto de este operador es que, tras ser ejecutado, el rover pasa a transportar el suministro y este ya no está en la base indicada. Además, a partir de la extensión 1, indicamos que el rover está llevando un suministro.

|              |  |       |
|--------------|--|-------|
| Parámetros   | Rover <i>rover</i> , Base <i>pos</i> , Suministro <i>sum</i>   |       |
| Precondición | $posicionRover(rover, pos) \wedge posicionRecurso(sum, pos) \wedge \neg entregado(sum)$  | Base  |
|              | $posicionRover(rover, pos) \wedge posicionRecurso(sum, pos) \wedge \neg entregado(sum) \wedge \neg llevaSuministro(rover) \wedge numPersonas(rover) = 0$ | Ext 1 |
| Efecto       | $transportaSuministro(rover, sum) \wedge \neg posicionRecurso(sum, pos)$   | Base  |

|  |  |       |
|--|--|-------|
|  | $\neg \text{posicionRecurso}(\text{sum}, \text{pos}) \wedge \text{transportaSuministro}(\text{rover}, \text{sum}) \wedge \text{llevaSuministro}(\text{rover})$ | Ext 1 |
|--|--|-------|

Tabla 2: Descripción del operador *cogerSuministro*

## Coger Persona

El operador *cogerPersona* es simétrico al de *cogerSuministro*, con la diferencia de que estamos recogiendo un recurso de subtipo *Persona* en vez de subtipo *Suministro*. Decidimos separar estos dos casos en dos operadores distintos, con tal de estar los operadores más orientados a objetos, permitiendo así realizar cambios más fácilmente en caso de querer añadir condiciones a las personas o los suministros.

Este operador se encarga de recoger a una persona de una base en un rover. Recibe como parámetros un rover, una base y una persona, y tiene como precondition que tanto el rover como la persona se encuentren en la base indicada, y la persona aún no haya sido asignada. Además, a partir de la extensión 1, el rover no puede llevar antes de recoger la persona ningún suministro, y como mucho una persona.

El efecto de este operador es que, tras ser ejecutado, el rover pasa a transportar la persona y esta ya no está en la base indicada. Además, a partir de la extensión 1, incrementamos el conteo de personas que transporta el rover.

|              |   |       |
|--------------|---|-------|
| Parámetros   | Rover <i>rover</i> , Base <i>pos</i> , Persona <i>per</i>   |       |
| Precondición | $\text{posicionRover}(\text{rover}, \text{pos}) \wedge \text{posicionRecurso}(\text{per}, \text{pos}) \wedge \neg \text{entregado}(\text{per})$   | Base  |
|              | $\text{posicionRover}(\text{rover}, \text{pos}) \wedge \text{posicionRecurso}(\text{per}, \text{pos}) \wedge \neg \text{entregado}(\text{per}) \wedge \neg \text{llevaSuministro}(\text{rover}) \wedge \text{numPersonas}(\text{rover}) \leq 1$ | Ext 1 |
| Efecto       | $\text{transportaPersona}(\text{rover}, \text{per}) \wedge \neg \text{posicionRecurso}(\text{per}, \text{pos})$   | Base  |
|              | $\text{transportaPersona}(\text{rover}, \text{per}) \wedge \neg \text{posicionRecurso}(\text{per}, \text{pos}) \wedge \text{increase}(\text{numPersonas}(\text{rover}), 1)$   | Ext 1 |

Tabla 3: Descripción del operador *cogerPersona*

## Dejar Suministro

Este operador sirve para dejar un suministro que lleva un rover en una base. Recibe como parámetros un rover, una base y un suministro y tiene como precondition que el rover esté en la base, exista una petición del suministro indicado a la base, y el rover actualmente transporte el suministro.

El efecto de este operador es que el rover deja de transportar el suministro, deja de haber una petición del suministro a la base, el suministro pasa a estar en la nueva base y se marca el recurso como entregado. A partir de la extensión 1, además, marcamos que el rover deja de tener un suministro cargado, y a partir de la extensión 3, incrementamos *sumaPrioridad* por la prioridad del pedido que acabamos de satisfacer.

|              |   |  |       |
|--------------|---|--|-------|
| Parámetros   | Rover <i>rover</i> , Base <i>pos</i> , Suministro <i>sum</i>                                  |  |       |
| Precondición | $posicionRover(rover, pos) \wedge peticion(sum, pos) \wedge transportaSuministro(rover, sum)$ |  |       |
| Efecto       | $\neg transportaSuministro(rover, sum) \wedge \neg peticion(sum, pos) \wedge$                 |  | Base  |
|              | $posicionRecurso(sum, pos) \wedge entregado(sum)$   |  |       |
|              | $\neg transportaSuministro(rover, sum) \wedge \neg peticion(sum, pos) \wedge$                 |  | Ext 1 |
|              | $posicionRecurso(sum, pos) \wedge entregado(sum) \wedge \neg llevaSuministro(rover)$          |  |       |
|              | $\neg transportaSuministro(rover, sum) \wedge \neg peticion(sum, pos) \wedge$                 |  | Ext 3 |
|              | $posicionRecurso(sum, pos) \wedge entregado(sum) \wedge \neg llevaSuministro(rover) \wedge$   |  |       |
|              | $increase(sumaPrioridad\ prioridad(sum, pos))$  |  |       |

Tabla 4: Descripción del operador *dejarSuministro*

## Dejar Persona

El operador *dejarPersona* es simétrico al de *dejarSuministro*, de la misma manera que lo era el de *cogerPersona* al de *cogerSuministro*. Es decir, realiza prácticamente los mismos cambios pero con un recurso de subtipo Persona esta vez. La justificación de esta separación se ha dado en la descripción del operador *cogerPersona*.

Este operador sirve para dejar una persona que lleva un rover en una base. Recibe como parámetros un rover, una base y una persona y tiene como precondition que el rover esté

en la base, exista una petición de la persona indicada a la base, y el rover actualmente transporte la persona.

El efecto de este operador es que el rover deja de transportar la persona, deja de haber una petición de la persona a la base, la persona pasa a estar en la nueva base y el recurso se marca como entregado. A partir de la extensión 1, además, decrementamos en uno el número de personas que transporta el rover, y a partir de la extensión 3, incrementamos *sumaPrioridad* por la prioridad del pedido que acabamos de satisfacer.

|              |  |
|--------------|--|
| Parámetros   | Rover <i>rover</i> , Base <i>pos</i> , Persona <i>per</i>  |
| Precondición | $posicionRover(rover, pos) \wedge peticion(per, pos) \wedge transportaPersona(rover, per)$   |
| Efecto       | $\neg transportaPersona(rover, per) \wedge \neg peticion(per, pos) \wedge$ Base<br>$posicionRecurso(per, pos) \wedge entregado(per)$<br><br>$\neg transportaPersona(rover, per) \wedge \neg peticion(per, pos) \wedge$ Ext 1<br>$posicionRecurso(per, pos) \wedge entregado(per) \wedge decrease(numPersonas(rover), 1)$<br><br>$\neg transportaPersona(rover, per) \wedge \neg peticion(per, pos) \wedge$ Ext 3<br>$posicionRecurso(per, pos) \wedge entregado(per) \wedge decrease(numPersonas(rover), 1) \wedge$<br>$increase(sumaPrioridad prioridad(per, pos))$ |

Tabla 5: Descripción del operador *dejarPersona*

# Juegos de prueba

## Prueba 1

Este juego de prueba es para la versión básica del proyecto.

### Introducción

Esta prueba se ha generado con el programa de generación aleatorio de casos. Cuenta con 22 recursos diferentes (10 personas y 12 suministros), 39 peticiones, 6 rovers y 12 bases conectadas entre ellas (7 asentamientos y 5 almacenes). El programa debería usar pocos rovers, ya que, al no tener limitación de carga, no es necesario movilizar muchos rovers para cargar con todo.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-basico)
  (:objects
    p0 p1 p2 p3 p4 p5 p6 p7 p8 p9 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 - suministro
    a0 a1 a2 a3 a4 a5 a6 - asentamiento
    al0 al1 al2 al3 al4 - almacen
    r0 r1 r2 r3 r4 r5 - rover
  )
  (:init
    (basesAdyacentes a0 a1)
    (basesAdyacentes a0 al4)
    (basesAdyacentes a1 a0)
    (basesAdyacentes a1 a4)
    (basesAdyacentes a2 a3)
    (basesAdyacentes a2 al4)
    (basesAdyacentes a3 al1)
    (basesAdyacentes a3 a2)
    (basesAdyacentes a3 a6)
    (basesAdyacentes a4 a1)
    (basesAdyacentes a4 al2)
    (basesAdyacentes a4 a6)
    (basesAdyacentes a5 a6)
    (basesAdyacentes a6 al3)
    (basesAdyacentes a6 a3)
    (basesAdyacentes a6 a4)
    (basesAdyacentes a6 a5)
    (basesAdyacentes al0 al1)
    (basesAdyacentes al1 a3)
    (basesAdyacentes al1 al0)
    (basesAdyacentes al2 al4)
    (basesAdyacentes al2 a4)
```

```
(basesAdyacentes a13 a6)
(basesAdyacentes a14 a0)
(basesAdyacentes a14 a12)
(basesAdyacentes a14 a2)
(posicionRecurso p0 a3)
(posicionRecurso p1 a6)
(posicionRecurso p2 a2)
(posicionRecurso p3 a1)
(posicionRecurso p4 a5)
(posicionRecurso p5 a2)
(posicionRecurso p6 a5)
(posicionRecurso p7 a4)
(posicionRecurso p8 a0)
(posicionRecurso p9 a0)
(posicionRecurso s0 a12)
(posicionRecurso s1 a10)
(posicionRecurso s2 a13)
(posicionRecurso s3 a14)
(posicionRecurso s4 a11)
(posicionRecurso s5 a11)
(posicionRecurso s6 a10)
(posicionRecurso s7 a14)
(posicionRecurso s8 a11)
(posicionRecurso s9 a11)
(posicionRecurso s10 a12)
(posicionRecurso s11 a13)
(peticion p0 a5)
(peticion p1 a3)
(peticion p2 a4)
(peticion p3 a4)
(peticion p4 a5)
(peticion p5 a6)
(peticion p6 a3)
(peticion p7 a4)
(peticion p8 a0)
(peticion p9 a3)
(peticion s0 a13)
(peticion s1 a10)
(peticion s2 a11)
(peticion s3 a13)
(peticion s4 a14)
(peticion s5 a10)
(peticion s6 a11)
(peticion s7 a12)
(peticion s8 a11)
(peticion s9 a12)
(peticion s10 a12)
(peticion s11 a10)
(peticion p3 a0)
(peticion s7 a10)
(peticion s10 a12)
(peticion s9 a14)
(peticion p4 a6)
(peticion s10 a11)
(peticion s6 a12)
(peticion p6 a5)
```

```

(peticion s3 a11)
(peticion s0 a10)
(peticion p7 a0)
(peticion s8 a10)
(peticion s8 a13)
(peticion p6 a3)
(peticion p5 a1)
(peticion s4 a10)
(peticion s11 a11)
(posicionRover r0 a10)
(posicionRover r1 a6)
(posicionRover r2 a10)
(posicionRover r3 a1)
(posicionRover r4 a10)
(posicionRover r5 a11)
)
(:goal (forall (?r - recurso) (entregado ?r)))
)

```

## Salida FF

```

ff: parsing domain file
domain 'DESPLAZAMIENTOS-BASICO' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

no metric specified. plan length assumed.

task contains conditional effects. turning off state domination.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on 1*g(s) + 5*h(s) where
metric is plan length

Cueing down from goal distance: 52 into depth [1]
51 [1]
50 [1]
49 [1]
47 [1]
46 [1]
45 [1]
43 [1]
42 [1]
40 [1]
36 [1]
34 [1]
31 [1]
30 [1]

```

|    |        |
|----|--------|
| 29 | [1]    |
| 28 | [1]    |
| 27 | [1]    |
| 26 | [1]    |
| 25 | [1]    |
| 24 | [1]    |
| 22 | [1]    |
| 21 | [1]    |
| 20 | [1]    |
| 19 | [1]    |
| 18 | [1][2] |
| 17 | [1]    |
| 16 | [1]    |
| 15 | [1]    |
| 14 | [1]    |
| 12 | [1]    |
| 11 | [1]    |
| 10 | [1][2] |
| 9  | [1]    |
| 7  | [1]    |
| 6  | [1][2] |
| 4  | [1]    |
| 3  | [1][2] |
| 1  | [1]    |
| 0  |        |

ff: found legal plan as follows

```

step  0: MOVER R1 A6 A5
      1: MOVER R1 A6 AL3
      2: COGER R1 AL3 S11
      3: COGER R4 AL0 S6
      4: COGER R1 AL3 S2
      5: MOVER R1 A6 A3
      6: MOVER R1 A6 A4
      7: MOVER R1 A3 AL1
      8: MOVER R1 A4 AL2
      9: MOVER R1 AL2 AL4
     10: MOVER R1 AL4 A0
     11: MOVER R1 AL4 A2
     12: COGER R1 AL2 S10
     13: COGER R1 AL1 S9
     14: COGER R1 AL4 S3
     15: COGER R4 AL0 S1
     16: COGER R1 A0 P8
     17: COGER R3 A1 P3
     18: COGER R1 A3 P0
     19: MOVER R1 AL4 AL2
     20: MOVER R1 A3 AL1
     21: MOVER R1 AL4 A0
     22: MOVER R3 A1 A0
     23: MOVER R1 A6 A5
     24: COGER R1 AL1 S8
     25: COGER R1 AL4 S7
     26: COGER R1 AL1 S5
     27: COGER R1 AL1 S4

```



```
28: MOVER R4 AL0 AL1
29: COGER R1 AL2 S0
30: MOVER R1 AL1 AL0
31: MOVER R4 AL1 AL0
32: COGER R1 A0 P9
33: COGER R1 A5 P6
34: COGER R1 A6 P1
35: MOVER R1 AL1 A3
36: COGER R1 A4 P7
37: COGER R1 A2 P2
38: MOVER R1 AL2 A4
39: COGER R1 A2 P5
40: COGER R1 A5 P4
41: MOVER R1 AL3 A6
```

```
time spent:  0.00 seconds instantiating 1740 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 309 facts and 288 actions
            0.00 seconds creating final representation with 301 relevant facts, 0
relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 220 states, to a max depth of 2
            0.00 seconds total time
```

## Resultado

El programa consigue ejecutarse de forma prácticamente instantánea y genera un plan correcto en tan solo 41 movimientos (cabe destacar que en el programa básico, la acción de mover también deja todos los paquetes que lleve si esa base es un destino posible. Tampoco se diferencia entre tipos de recurso). Como hemos comentado en la introducción del juego de prueba y como se puede apreciar en los resultados, prácticamente solo se usa un rover para mover todos los recursos. El resto de rovers solo se mueven para trasladar paquetes que inicialmente están en la misma base hasta posiciones muy cercanas.

## Prueba 2

Este juego de prueba es para la versión básica del proyecto.

### Introducción

Este juego también se ha creado con el generador de juegos de prueba y contiene 13 recursos (5 personas y 8 suministros), 14 bases (7 asentamientos y 7 almacenes), 4 rovers y 27 peticiones. Los resultados deberían ser muy parecidos al test anterior, solo que debería generar una solución en menos movimientos, ya que el número de recursos es menor y hay bastantes más adyacencias de bases.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-basico)
  (:objects
    p0 p1 p2 p3 p4 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 - suministro
    a0 a1 a2 a3 a4 a5 a6 - asentamiento
    al0 al1 al2 al3 al4 al5 al6 - almacen
    r0 r1 r2 r3 - rover
  )
  (:init
    (basesAdyacentes a0 al2)
    (basesAdyacentes a1 al1)
    (basesAdyacentes a1 al4)
    (basesAdyacentes a1 a5)
    (basesAdyacentes a2 al2)
    (basesAdyacentes a2 al3)
    (basesAdyacentes a2 al4)
    (basesAdyacentes a2 al5)
    (basesAdyacentes a3 al1)
    (basesAdyacentes a3 al0)
    (basesAdyacentes a4 al3)
    (basesAdyacentes a4 al6)
    (basesAdyacentes a4 al0)
    (basesAdyacentes a5 a1)
    (basesAdyacentes a5 a6)
    (basesAdyacentes a6 al6)
    (basesAdyacentes a6 a5)
    (basesAdyacentes al0 a3)
    (basesAdyacentes al0 a4)
    (basesAdyacentes al0 al5)
    (basesAdyacentes al1 a3)
    (basesAdyacentes al1 a1)
    (basesAdyacentes al1 al4)
    (basesAdyacentes al2 a0)
    (basesAdyacentes al2 a2)
    (basesAdyacentes al3 a2)
    (basesAdyacentes al3 a4)
```

```

(basesAdyacentes a14 a11)
(basesAdyacentes a14 a1)
(basesAdyacentes a14 a2)
(basesAdyacentes a15 a2)
(basesAdyacentes a15 a10)
(basesAdyacentes a16 a4)
(basesAdyacentes a16 a6)
(posicionRecurso p0 a0)
(posicionRecurso p1 a6)
(posicionRecurso p2 a3)
(posicionRecurso p3 a4)
(posicionRecurso p4 a5)
(posicionRecurso s0 a13)
(posicionRecurso s1 a10)
(posicionRecurso s2 a11)
(posicionRecurso s3 a12)
(posicionRecurso s4 a16)
(posicionRecurso s5 a10)
(posicionRecurso s6 a15)
(posicionRecurso s7 a10)
(peticion p0 a1)
(peticion p1 a4)
(peticion p2 a2)
(peticion p3 a4)
(peticion p4 a6)
(peticion s0 a16)
(peticion s1 a12)
(peticion s2 a15)
(peticion s3 a14)
(peticion s4 a15)
(peticion s5 a14)
(peticion s6 a12)
(peticion s7 a11)
(peticion p1 a6)
(peticion p1 a4)
(peticion s2 a16)
(peticion s2 a11)
(peticion p4 a3)
(peticion s7 a10)
(peticion p3 a2)
(peticion p3 a1)
(peticion p1 a3)
(peticion p3 a1)
(peticion p0 a6)
(peticion p4 a2)
(peticion s1 a16)
(peticion s4 a15)
(posicionRover r0 a2)
(posicionRover r1 a4)
(posicionRover r2 a16)
(posicionRover r3 a1)
)
(:goal (forall (?r - recurso) (entregado ?r)))
)

```

## Salida FF

```
ff: parsing domain file
domain 'DESPLAZAMIENTOS-BASICO' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

no metric specified. plan length assumed.

task contains conditional effects. turning off state domination.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on  $1*g(s) + 5*h(s)$  where
metric is plan length

Cueing down from goal distance: 38 into depth [1]
37 [1]
35 [1]
34 [1]
32 [1]
31 [1]
30 [1]
29 [1]
28 [1]
26 [1]
25 [1]
23 [1]
22 [1]
21 [1]
20 [1]
19 [1]
18 [1]
17 [1]
16 [1]
15 [1]
14 [1]
13 [1]
12 [1]
11 [1]
10 [1]
9 [1][2]
7 [1]
6 [1][2]
4 [1]
3 [1][2]
1 [1]
0

ff: found legal plan as follows
```

```

step    0: MOVER R0 A2 AL5
        1: MOVER R2 AL6 A4
        2: MOVER R0 AL5 AL0
        3: MOVER R0 A2 AL2
        4: MOVER R3 A1 AL1
        5: MOVER R2 A4 AL3
        6: MOVER R3 A1 A5
        7: MOVER R0 AL0 A3
        8: MOVER R3 A5 A6
        9: MOVER R0 AL2 A0
       10: COGER R0 AL0 S7
       11: COGER R2 AL6 S4
       12: MOVER R2 AL3 A2
       13: COGER R3 AL1 S2
       14: COGER R2 AL3 S0
       15: COGER R2 A4 P3
       16: COGER R0 A3 P2
       17: COGER R0 A0 P0
       18: MOVER R0 A3 AL1
       19: MOVER R2 A2 AL5
       20: MOVER R3 A1 AL1
       21: MOVER R2 A4 AL6
       22: MOVER R2 AL5 A2
       23: MOVER R0 AL2 A2
       24: MOVER R0 AL1 A1
       25: COGER R0 AL5 S6
       26: COGER R0 AL0 S1
       27: MOVER R0 A0 AL2
       28: COGER R0 AL0 S5
       29: COGER R0 AL2 S3
       30: MOVER R0 A1 AL4
       31: COGER R3 A5 P4
       32: COGER R3 A6 P1
       33: MOVER R3 A5 A6

time spent:  0.00 seconds instantiating 864 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 174 facts and 188 actions
            0.00 seconds creating final representation with 167 relevant facts, 0
relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 124 states, to a max depth of 2
            0.00 seconds total time

```

## Resultado

El resultado es el esperado. El programa se sigue ejecutando en muy poco tiempo y en menos movimientos que en la prueba anterior. En este caso sí que se reparte más el uso de los rovers, probablemente debido a que los paquetes se han situado en bases muy cercanas a su destino (por la gran cantidad de adyacencias) así que los rovers se han podido repartir mejor los itinerarios para no superponer sus trayectos.

## Prueba 3

Este juego de prueba es para la extensión 1 del proyecto.

### Introducción

Este juego, que también se ha generado con nuestro programa de generación de juegos de prueba, consta de 19 recursos (9 personas y 10 suministros), 12 bases (7 asentamientos y 5 almacenes), 8 rovers y 29 peticiones. En esta prueba se usa la extensión 1, que introduce una capacidad máxima en los rovers, por lo que se debería notar que se usan muchos más rovers que en la versión básica.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-ext1)
  (:objects
    p0 p1 p2 p3 p4 p5 p6 p7 p8 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 - suministro
    a0 a1 a2 a3 a4 a5 a6 - asentamiento
    al0 al1 al2 al3 al4 - almacen
    r0 r1 r2 r3 r4 r5 r6 r7 - rover
  )
  (:init
    (basesAdyacentes a0 a2)
    (basesAdyacentes a0 al4)
    (basesAdyacentes a0 a5)
    (basesAdyacentes a0 al0)
    (basesAdyacentes a1 a2)
    (basesAdyacentes a1 a5)
    (basesAdyacentes a1 a6)
    (basesAdyacentes a1 al0)
    (basesAdyacentes a1 al1)
    (basesAdyacentes a1 al2)
    (basesAdyacentes a1 al3)
    (basesAdyacentes a1 al4)
    (basesAdyacentes a2 a0)
    (basesAdyacentes a2 a1)
    (basesAdyacentes a3 al4)
    (basesAdyacentes a4 al3)
    (basesAdyacentes a5 a0)
    (basesAdyacentes a5 a1)
    (basesAdyacentes a5 al1)
    (basesAdyacentes a6 a1)
    (basesAdyacentes a6 al3)
    (basesAdyacentes a6 al4)
    (basesAdyacentes a6 al0)
    (basesAdyacentes al0 al1)
    (basesAdyacentes al0 a1)
    (basesAdyacentes al0 a0)
```

```
(basesAdyacentes a10 a6)
(basesAdyacentes a11 a1)
(basesAdyacentes a11 a13)
(basesAdyacentes a11 a5)
(basesAdyacentes a11 a10)
(basesAdyacentes a12 a1)
(basesAdyacentes a13 a11)
(basesAdyacentes a13 a1)
(basesAdyacentes a13 a4)
(basesAdyacentes a13 a6)
(basesAdyacentes a14 a0)
(basesAdyacentes a14 a1)
(basesAdyacentes a14 a3)
(basesAdyacentes a14 a6)
(posicionRecurso p0 a1)
(posicionRecurso p1 a1)
(posicionRecurso p2 a2)
(posicionRecurso p3 a3)
(posicionRecurso p4 a1)
(posicionRecurso p5 a5)
(posicionRecurso p6 a3)
(posicionRecurso p7 a5)
(posicionRecurso p8 a3)
(posicionRecurso s0 a14)
(posicionRecurso s1 a11)
(posicionRecurso s2 a10)
(posicionRecurso s3 a12)
(posicionRecurso s4 a12)
(posicionRecurso s5 a12)
(posicionRecurso s6 a11)
(posicionRecurso s7 a14)
(posicionRecurso s8 a14)
(posicionRecurso s9 a11)
(petition p0 a6)
(petition p1 a3)
(petition p2 a4)
(petition p3 a5)
(petition p4 a6)
(petition p5 a4)
(petition p6 a5)
(petition p7 a6)
(petition p8 a1)
(petition s0 a12)
(petition s1 a13)
(petition s2 a12)
(petition s3 a13)
(petition s4 a10)
(petition s5 a13)
(petition s6 a12)
(petition s7 a10)
(petition s8 a13)
(petition s9 a12)
(petition p6 a3)
(petition s9 a11)
(petition p1 a1)
(petition s5 a14)
```

```

(peticion p1 a0)
(peticion s0 a10)
(peticion p1 a2)
(peticion p4 a1)
(peticion p1 a1)
(peticion p6 a6)
(posicionRover r0 a2) (= (numPersonas r0) 0)
(posicionRover r1 a10) (= (numPersonas r1) 0)
(posicionRover r2 a1) (= (numPersonas r2) 0)
(posicionRover r3 a2) (= (numPersonas r3) 0)
(posicionRover r4 a0) (= (numPersonas r4) 0)
(posicionRover r5 a10) (= (numPersonas r5) 0)
(posicionRover r6 a1) (= (numPersonas r6) 0)
(posicionRover r7 a14) (= (numPersonas r7) 0)
)
(:goal (forall (?r - recurso) (entregado ?r)))
)

```

## Salida FF

```

ff: parsing domain file
domain 'DESPLAZAMIENTOS-EXT1' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

no metric specified. plan length assumed.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on 1*g(s) + 5*h(s) where
metric is plan length

Cueing down from goal distance: 55 into depth [1]
50 [1]
49 [1]
48 [1]
47 [1]
46 [1][2]
45 [1]
44 [1][2]
43 [1][2]
42 [1][2]
41 [1][2][3]
40 [1]
39 [1]
38 [1][2][3][4][5]
37 [1]
36 [1][2][3]
35 [1][2][3][4]

```



|    |              |
|----|--------------|
| 34 | [1][2]       |
| 33 | [1][2][3][4] |
| 32 | [1][2]       |
| 31 | [1][2][3][4] |
| 30 | [1][2]       |
| 29 | [1][2][3][4] |
| 27 | [1]          |
| 26 | [1][2][3][4] |
| 25 | [1][2][3]    |
| 24 | [1][2][3]    |
| 23 | [1][2][3]    |
| 22 | [1][2][3][4] |
| 21 | [1][2][3]    |
| 20 | [1]          |
| 19 | [1][2][3]    |
| 18 | [1][2][3]    |
| 17 | [1][2][3]    |
| 15 | [1][2][3]    |
| 14 | [1][2][3]    |
| 13 | [1][2][3]    |
| 12 | [1]          |
| 11 | [1]          |
| 10 | [1][2][3][4] |
| 9  | [1]          |
| 8  | [1]          |
| 7  | [1]          |
| 6  | [1]          |
| 5  | [1]          |
| 3  | [1]          |
| 2  | [1]          |
| 1  | [1]          |
| 0  |              |

ff: found legal plan as follows

```

step  0: MOVER R7 AL4 A0
      1: COGERPERSONA R6 A1 P4
      2: DEJARPERSONA R6 A1 P4
      3: COGERPERSONA R6 A1 P1
      4: DEJARPERSONA R6 A1 P1
      5: MOVER R7 A0 AL4
      6: COGERSUMINISTRO R7 AL4 S8
      7: MOVER R7 AL4 A1
      8: MOVER R7 A1 AL3
      9: DEJARSUMINISTRO R7 AL3 S8
     10: MOVER R7 AL3 AL1
     11: COGERSUMINISTRO R7 AL1 S1
     12: MOVER R7 AL1 AL3
     13: DEJARSUMINISTRO R7 AL3 S1
     14: MOVER R6 A1 AL1
     15: MOVER R4 A0 AL4
     16: MOVER R4 AL4 A1
     17: COGERSUMINISTRO R6 AL1 S9
     18: DEJARSUMINISTRO R6 AL1 S9
     19: MOVER R4 A1 AL1
     20: MOVER R5 AL0 A1

```

21: COGERSUMINISTRO R4 AL1 S6  
22: MOVER R4 AL1 A1  
23: MOVER R6 AL1 AL3  
24: MOVER R4 A1 AL2  
25: MOVER R5 A1 A2  
26: DEJARSUMINISTRO R4 AL2 S6  
27: MOVER R4 AL2 A1  
28: MOVER R4 A1 AL4  
29: COGERSUMINISTRO R4 AL4 S7  
30: MOVER R4 AL4 A1  
31: MOVER R4 A1 AL0  
32: DEJARSUMINISTRO R4 AL0 S7  
33: MOVER R4 AL0 A1  
34: MOVER R4 A1 AL4  
35: COGERSUMINISTRO R4 AL4 S0  
36: MOVER R4 AL4 A1  
37: MOVER R4 A1 AL2  
38: DEJARSUMINISTRO R4 AL2 S0  
39: MOVER R4 AL2 A1  
40: MOVER R4 A1 AL0  
41: COGERSUMINISTRO R4 AL0 S2  
42: MOVER R4 AL0 A1  
43: MOVER R4 A1 AL2  
44: DEJARSUMINISTRO R4 AL2 S2  
45: COGERSUMINISTRO R4 AL2 S4  
46: MOVER R4 AL2 A1  
47: MOVER R4 A1 AL0  
48: DEJARSUMINISTRO R4 AL0 S4  
49: MOVER R2 A1 A2  
50: MOVER R6 AL3 AL1  
51: MOVER R7 AL3 A1  
52: COGERPERSONA R7 A1 P0  
53: MOVER R7 A1 A6  
54: DEJARPERSONA R7 A6 P0  
55: MOVER R7 A6 A1  
56: MOVER R7 A1 A5  
57: COGERPERSONA R7 A5 P7  
58: MOVER R7 A5 A1  
59: MOVER R7 A1 A6  
60: DEJARPERSONA R7 A6 P7  
61: MOVER R7 A6 A1  
62: MOVER R7 A1 A2  
63: MOVER R6 AL1 AL3  
64: COGERPERSONA R7 A2 P2  
65: MOVER R7 A2 A1  
66: MOVER R6 AL3 A1  
67: MOVER R7 A1 AL3  
68: MOVER R7 AL3 A4  
69: DEJARPERSONA R7 A4 P2  
70: MOVER R6 A1 AL2  
71: COGERSUMINISTRO R6 AL2 S5  
72: MOVER R6 AL2 A1  
73: MOVER R6 A1 AL4  
74: DEJARSUMINISTRO R6 AL4 S5  
75: MOVER R6 AL4 A1  
76: MOVER R6 A1 AL2

```

77: COGERSUMINISTRO R6 AL2 S3
78: MOVER R6 AL2 A1
79: MOVER R6 A1 AL3
80: DEJARSUMINISTRO R6 AL3 S3
81: MOVER R6 AL3 AL1
82: MOVER R6 AL1 A5
83: COGERPERSONA R6 A5 P5
84: MOVER R6 A5 AL1
85: MOVER R6 AL1 AL3
86: MOVER R6 AL3 A4
87: MOVER R5 A2 A0
88: DEJARPERSONA R6 A4 P5
89: MOVER R5 A0 AL4
90: MOVER R5 AL4 A3
91: COGERPERSONA R5 A3 P6
92: COGERPERSONA R5 A3 P8
93: DEJARPERSONA R5 A3 P6
94: COGERPERSONA R5 A3 P3
95: MOVER R5 A3 AL4
96: MOVER R5 AL4 A1
97: DEJARPERSONA R5 A1 P8
98: MOVER R5 A1 A5
99: DEJARPERSONA R5 A5 P3

```

```

time spent:  0.00 seconds instantiating 3968 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 374 facts and 888 actions
            0.00 seconds creating final representation with 373 relevant facts, 16
relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            1.49 seconds searching, evaluating 18328 states, to a max depth of 5
            1.49 seconds total time

```

## Resultado

Podemos observar que se ha repartido de forma más equitativa la distribución de paquetes para cada rover, pero siguen habiendo rovers sin uso alguno (en este caso, el R0, R1 y R2 prácticamente no se han utilizado). La razón es que estos rovers se han situado inicialmente en la misma posición que otros, por lo que no han encontrado ningún pedido que entregar. El programa ha tardado más que los dos primeros tests, ya que se trata de una extensión con más limitaciones, y también se ha generado un plan con más acciones, pues no se pueden transportar recursos ilimitados (y además separamos las acciones dejar y coger para cada recurso)

## Prueba 4

Este juego de prueba es para la extensión 1 del proyecto.

### Introducción

En este juego de pruebas deberíamos encontrar resultados muy parecidos al anterior, ya que la cantidad de objetos es muy parecida (7 personas, 9 suministros, 6 asentamientos, 6 almacenes, 6 rovers y 26 peticiones). La principal diferencia es que en este test hay muchas más adyacencias, con lo que deberíamos notar que se requieren menos acciones para entregar todos los recursos porque no serán necesarias tantas acciones de “mover”.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-ext1)
  (:objects
    p0 p1 p2 p3 p4 p5 p6 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 s8 - suministro
    a0 a1 a2 a3 a4 a5 - asentamiento
    al0 al1 al2 al3 al4 al5 - almacen
    r0 r1 r2 r3 r4 r5 - rover
  )
  (:init
    (basesAdyacentes a0 al5)
    (basesAdyacentes a0 al3)
    (basesAdyacentes a0 al4)
    (basesAdyacentes a0 a2)
    (basesAdyacentes a1 al2)
    (basesAdyacentes a1 a2)
    (basesAdyacentes a1 a4)
    (basesAdyacentes a2 a0)
    (basesAdyacentes a2 a1)
    (basesAdyacentes a2 al0)
    (basesAdyacentes a2 al1)
    (basesAdyacentes a2 al2)
    (basesAdyacentes a2 al5)
    (basesAdyacentes a3 al2)
    (basesAdyacentes a3 al4)
    (basesAdyacentes a3 a4)
    (basesAdyacentes a3 al1)
    (basesAdyacentes a4 a1)
    (basesAdyacentes a4 al4)
    (basesAdyacentes a4 a3)
    (basesAdyacentes a4 al1)
    (basesAdyacentes a5 al2)
    (basesAdyacentes al0 a2)
    (basesAdyacentes al0 al1)
    (basesAdyacentes al0 al2)
    (basesAdyacentes al0 al3)
    (basesAdyacentes al0 al4)
```

(basesAdyacentes a11 a2)  
(basesAdyacentes a11 a3)  
(basesAdyacentes a11 a4)  
(basesAdyacentes a11 a10)  
(basesAdyacentes a11 a12)  
(basesAdyacentes a11 a13)  
(basesAdyacentes a11 a14)  
(basesAdyacentes a11 a15)  
(basesAdyacentes a12 a1)  
(basesAdyacentes a12 a2)  
(basesAdyacentes a12 a3)  
(basesAdyacentes a12 a5)  
(basesAdyacentes a12 a10)  
(basesAdyacentes a12 a11)  
(basesAdyacentes a12 a14)  
(basesAdyacentes a12 a15)  
(basesAdyacentes a13 a0)  
(basesAdyacentes a13 a15)  
(basesAdyacentes a13 a10)  
(basesAdyacentes a13 a11)  
(basesAdyacentes a14 a0)  
(basesAdyacentes a14 a3)  
(basesAdyacentes a14 a4)  
(basesAdyacentes a14 a10)  
(basesAdyacentes a14 a11)  
(basesAdyacentes a14 a12)  
(basesAdyacentes a14 a15)  
(basesAdyacentes a15 a0)  
(basesAdyacentes a15 a2)  
(basesAdyacentes a15 a11)  
(basesAdyacentes a15 a12)  
(basesAdyacentes a15 a13)  
(basesAdyacentes a15 a14)  
(posicionRecurso p0 a2)  
(posicionRecurso p1 a1)  
(posicionRecurso p2 a4)  
(posicionRecurso p3 a1)  
(posicionRecurso p4 a4)  
(posicionRecurso p5 a2)  
(posicionRecurso p6 a3)  
(posicionRecurso s0 a13)  
(posicionRecurso s1 a13)  
(posicionRecurso s2 a12)  
(posicionRecurso s3 a15)  
(posicionRecurso s4 a15)  
(posicionRecurso s5 a10)  
(posicionRecurso s6 a10)  
(posicionRecurso s7 a11)  
(posicionRecurso s8 a14)  
(petition p0 a4)  
(petition p1 a4)  
(petition p2 a2)  
(petition p3 a2)  
(petition p4 a1)  
(petition p5 a3)  
(petition p6 a4)

```

(peticion s0 a11)
(peticion s1 a11)
(peticion s2 a15)
(peticion s3 a10)
(peticion s4 a13)
(peticion s5 a12)
(peticion s6 a14)
(peticion s7 a12)
(peticion s8 a12)
(peticion s1 a10)
(peticion s3 a14)
(peticion p4 a5)
(peticion p2 a4)
(peticion s7 a11)
(peticion p5 a3)
(peticion s0 a13)
(peticion s4 a13)
(peticion s3 a12)
(peticion p4 a0)
(posicionRover r0 a1) (= (numPersonas r0) 0)
(posicionRover r1 a12) (= (numPersonas r1) 0)
(posicionRover r2 a5) (= (numPersonas r2) 0)
(posicionRover r3 a10) (= (numPersonas r3) 0)
(posicionRover r4 a14) (= (numPersonas r4) 0)
(posicionRover r5 a14) (= (numPersonas r5) 0)
)
(:goal (forall (?r - recurso) (entregado ?r)))
)

```

## Salida FF

```

ff: parsing domain file
domain 'DESPLAZAMIENTOS-EXT1' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

no metric specified. plan length assumed.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on 1*g(s) + 5*h(s) where
metric is plan length

Cueing down from goal distance: 48 into depth [1]
46 [1]
45 [1]
43 [1]
42 [1]
41 [1]
40 [1][2]
39 [1]
38 [1]

```

|    |        |
|----|--------|
| 37 | [1]    |
| 36 | [1]    |
| 35 | [1]    |
| 34 | [1][2] |
| 33 | [1]    |
| 32 | [1]    |
| 31 | [1]    |
| 30 | [1][2] |
| 29 | [1]    |
| 28 | [1]    |
| 27 | [1]    |
| 26 | [1]    |
| 25 | [1]    |
| 24 | [1]    |
| 23 | [1]    |
| 22 | [1][2] |
| 21 | [1]    |
| 20 | [1]    |
| 19 | [1]    |
| 18 | [1]    |
| 17 | [1]    |
| 16 | [1]    |
| 15 | [1]    |
| 14 | [1][2] |
| 13 | [1][2] |
| 12 | [1]    |
| 11 | [1]    |
| 10 | [1]    |
| 9  | [1][2] |
| 8  | [1]    |
| 7  | [1]    |
| 6  | [1]    |
| 5  | [1]    |
| 4  | [1][2] |
| 3  | [1]    |
| 2  | [1]    |
| 1  | [1]    |
| 0  |        |

ff: found legal plan as follows

```

step  0: MOVER R5 AL4 A3
      1: MOVER R4 AL4 AL2
      2: COGERPERSONA R5 A3 P6
      3: COGERPERSONA R0 A1 P3
      4: MOVER R3 AL0 AL3
      5: COGERSUMINISTRO R3 AL3 S1
      6: MOVER R4 AL2 AL5
      7: COGERSUMINISTRO R4 AL5 S4
      8: MOVER R4 AL5 AL3
      9: DEJARSUMINISTRO R4 AL3 S4
     10: COGERSUMINISTRO R4 AL3 S0
     11: MOVER R1 AL2 AL4
     12: DEJARSUMINISTRO R4 AL3 S0
     13: MOVER R5 A3 A4
     14: COGERPERSONA R5 A4 P4

```

```

15: DEJARPERSONA R5 A4 P6
16: COGERPERSONA R5 A4 P2
17: DEJARPERSONA R5 A4 P2
18: MOVER R0 A1 A2
19: COGERPERSONA R0 A2 P5
20: DEJARPERSONA R0 A2 P3
21: COGERPERSONA R0 A2 P0
22: MOVER R0 A2 AL1
23: MOVER R0 AL1 A3
24: DEJARPERSONA R0 A3 P5
25: MOVER R0 A3 A4
26: DEJARPERSONA R0 A4 P0
27: COGERSUMINISTRO R1 AL4 S8
28: MOVER R1 AL4 AL2
29: DEJARSUMINISTRO R1 AL2 S8
30: MOVER R3 AL3 AL1
31: DEJARSUMINISTRO R3 AL1 S1
32: COGERSUMINISTRO R3 AL1 S7
33: DEJARSUMINISTRO R3 AL1 S7
34: MOVER R3 AL1 AL0
35: COGERSUMINISTRO R3 AL0 S6
36: MOVER R3 AL0 AL4
37: DEJARSUMINISTRO R3 AL4 S6
38: MOVER R1 AL2 AL0
39: MOVER R3 AL4 AL2
40: COGERSUMINISTRO R1 AL0 S5
41: MOVER R1 AL0 AL2
42: DEJARSUMINISTRO R1 AL2 S5
43: COGERSUMINISTRO R3 AL2 S2
44: MOVER R3 AL2 AL5
45: DEJARSUMINISTRO R3 AL5 S2
46: COGERSUMINISTRO R3 AL5 S3
47: MOVER R3 AL5 AL4
48: DEJARSUMINISTRO R3 AL4 S3
49: MOVER R5 A4 A1
50: COGERPERSONA R5 A1 P1
51: DEJARPERSONA R5 A1 P4
52: MOVER R5 A1 A4
53: DEJARPERSONA R5 A4 P1

```

```

time spent:  0.00 seconds instantiating 2664 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 275 facts and 726 actions
            0.00 seconds creating final representation with 273 relevant facts, 12
relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.02 seconds searching, evaluating 419 states, to a max depth of 2
            0.02 seconds total time

```

## Resultado

Como esperábamos, el programa ha encontrado un plan de 53 pasos y ejecuta muy pocas veces la acción “mover”. También se ha encontrado la solución en mucho menos tiempo



gracias a tener un grafo de bases prácticamente completo y facilitar la movilidad de los rovers. La distribución de recursos en los rovers ha sido bastante equilibrada, lo cual es un resultado lógico teniendo en cuenta el límite de carga de cada rover y que todos los rovers recursos están a poca distancia de un destino (gracias a la completitud del grafo de bases)

## Prueba 5

Este juego de prueba es para la primera versión de la extensión 2 del proyecto.

### Introducción

Este juego también se ha creado con el generador de juegos de prueba y contiene 20 recursos (10 personas y 10 suministros), 15 bases (9 asentamientos y 6 almacenes), 4 rovers y 37 peticiones. En los resultados deberíamos ver que los rovers se reparten más el trabajo, ya que tienen una limitación de combustible y no es posible que una baja cantidad de rovers llegue a todos lados. De todas maneras, en esta versión de la extensión no se busca minimizar la cantidad de combustible gastado, con lo que no se deberían apreciar tanto las diferencias. Los rovers tienen inicialmente 18, 10, 19 y 26 unidades de combustible respectivamente.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-ext2)
  (:objects
    p0 p1 p2 p3 p4 p5 p6 p7 p8 p9 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 - suministro
    a0 a1 a2 a3 a4 a5 a6 a7 a8 - asentamiento
    al0 al1 al2 al3 al4 al5 - almacen
    r0 r1 r2 r3 - rover
  )
  (:init
    (basesAdyacentes a0 a1)
    (basesAdyacentes a0 a2)
    (basesAdyacentes a0 al5)
    (basesAdyacentes a1 a0)
    (basesAdyacentes a2 a0)
    (basesAdyacentes a3 a8)
    (basesAdyacentes a3 al0)
    (basesAdyacentes a3 a5)
    (basesAdyacentes a4 a5)
    (basesAdyacentes a4 a7)
    (basesAdyacentes a5 a3)
    (basesAdyacentes a5 a4)
    (basesAdyacentes a5 a6)
```

(basesAdyacentes a5 a13)  
(basesAdyacentes a5 a14)  
(basesAdyacentes a6 a12)  
(basesAdyacentes a6 a5)  
(basesAdyacentes a6 a7)  
(basesAdyacentes a7 a8)  
(basesAdyacentes a7 a4)  
(basesAdyacentes a7 a6)  
(basesAdyacentes a7 a15)  
(basesAdyacentes a8 a3)  
(basesAdyacentes a8 a7)  
(basesAdyacentes a10 a3)  
(basesAdyacentes a10 a14)  
(basesAdyacentes a11 a13)  
(basesAdyacentes a12 a6)  
(basesAdyacentes a13 a11)  
(basesAdyacentes a13 a14)  
(basesAdyacentes a13 a5)  
(basesAdyacentes a14 a10)  
(basesAdyacentes a14 a13)  
(basesAdyacentes a14 a5)  
(basesAdyacentes a15 a0)  
(basesAdyacentes a15 a7)  
(posicionRecurso p0 a4)  
(posicionRecurso p1 a8)  
(posicionRecurso p2 a7)  
(posicionRecurso p3 a2)  
(posicionRecurso p4 a7)  
(posicionRecurso p5 a3)  
(posicionRecurso p6 a1)  
(posicionRecurso p7 a4)  
(posicionRecurso p8 a7)  
(posicionRecurso p9 a5)  
(posicionRecurso s0 a14)  
(posicionRecurso s1 a13)  
(posicionRecurso s2 a15)  
(posicionRecurso s3 a14)  
(posicionRecurso s4 a10)  
(posicionRecurso s5 a13)  
(posicionRecurso s6 a15)  
(posicionRecurso s7 a13)  
(posicionRecurso s8 a10)  
(posicionRecurso s9 a14)  
(peticion p0 a3)  
(peticion p1 a7)  
(peticion p2 a8)  
(peticion p3 a4)  
(peticion p4 a8)  
(peticion p5 a4)  
(peticion p6 a5)  
(peticion p7 a2)  
(peticion p8 a4)  
(peticion p9 a6)  
(peticion s0 a13)  
(peticion s1 a11)  
(peticion s2 a11)

```

(peticion s3 a13)
(peticion s4 a10)
(peticion s5 a11)
(peticion s6 a12)
(peticion s7 a10)
(peticion s8 a11)
(peticion s9 a11)
(peticion p0 a3)
(peticion s2 a13)
(peticion s0 a13)
(peticion p0 a8)
(peticion p6 a8)
(peticion p2 a8)
(peticion p8 a2)
(peticion p9 a2)
(peticion p9 a1)
(peticion p4 a8)
(peticion p0 a0)
(peticion s2 a12)
(peticion p9 a4)
(peticion p5 a0)
(peticion s0 a14)
(peticion p1 a3)
(peticion s2 a11)
(posicionRover r0 a8) (= (numPersonas r0) 0) (= (combustible r0) 18)
(posicionRover r1 a11) (= (numPersonas r1) 0) (= (combustible r1) 10)
(posicionRover r2 a3) (= (numPersonas r2) 0) (= (combustible r2) 19)
(posicionRover r3 a12) (= (numPersonas r3) 0) (= (combustible r3) 26)
(= (sumaCombustible) 73)
)
(:goal (forall (?r - recurso) (entregado ?r)))
)

```

## Salida FF

```

ff: parsing domain file
domain 'DESPLAZAMIENTOS-EXT2' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

no metric specified. plan length assumed.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on 1*g(s) + 5*h(s) where
metric is plan length

Cueing down from goal distance: 59 into depth [1]
                                57 [1]
                                56 [1]
                                55 [1]

```

|    |                 |
|----|-----------------|
| 54 | [1]             |
| 53 | [1]             |
| 52 | [1][2]          |
| 51 | [1]             |
| 50 | [1][2]          |
| 49 | [1][2]          |
| 48 | [1]             |
| 46 | [1][2][3]       |
| 45 | [1]             |
| 44 | [1][2][3]       |
| 43 | [1]             |
| 42 | [1][2][3]       |
| 41 | [1]             |
| 40 | [1][2]          |
| 38 | [1][2]          |
| 37 | [1]             |
| 36 | [1]             |
| 35 | [1][2][3]       |
| 34 | [1]             |
| 33 | [1][2][3]       |
| 32 | [1][2][3]       |
| 31 | [1][2][3]       |
| 30 | [1][2][3][4]    |
| 29 | [1]             |
| 28 | [1][2][3][4]    |
| 27 | [1][2]          |
| 26 | [1]             |
| 25 | [1]             |
| 24 | [1][2][3][4]    |
| 23 | [1][2]          |
| 22 | [1][2]          |
| 21 | [1][2][3]       |
| 20 | [1]             |
| 19 | [1]             |
| 18 | [1]             |
| 17 | [1]             |
| 16 | [1]             |
| 15 | [1][2][3][4][5] |
| 14 | [1][2][3][4]    |
| 13 | [1]             |
| 12 | [1][2]          |
| 11 | [1][2][3][4][5] |
| 10 | [1]             |
| 9  | [1][2]          |
| 8  | [1]             |
| 7  | [1]             |
| 6  | [1]             |
| 5  | [1]             |
| 4  | [1]             |
| 3  | [1]             |
| 2  | [1]             |
| 1  | [1]             |
| 0  |                 |

ff: found legal plan as follows

```

step  0: MOVER R0 A8 A3
      1: MOVER R1 AL1 AL3
      2: MOVER R1 AL3 AL4
      3: COGERSUMINISTRO R1 AL4 S0
      4: DEJARSUMINISTRO R1 AL4 S0
      5: COGERPERSONA R2 A3 P5
      6: MOVER R3 AL2 A6
      7: MOVER R3 A6 A7
      8: COGERPERSONA R3 A7 P8
      9: MOVER R2 A3 A5
     10: MOVER R2 A5 A6
     11: MOVER R2 A6 A5
     12: COGERPERSONA R2 A5 P9
     13: MOVER R2 A5 A4
     14: MOVER R1 AL4 AL3
     15: MOVER R0 A3 AL0
     16: COGERSUMINISTRO R0 AL0 S4
     17: DEJARSUMINISTRO R0 AL0 S4
     18: MOVER R1 AL3 AL4
     19: COGERSUMINISTRO R1 AL4 S3
     20: MOVER R1 AL4 AL3
     21: DEJARSUMINISTRO R1 AL3 S3
     22: COGERPERSONA R3 A7 P2
     23: DEJARPERSONA R2 A4 P9
     24: COGERPERSONA R2 A4 P0
     25: DEJARPERSONA R2 A4 P5
     26: MOVER R2 A4 A7
     27: COGERPERSONA R2 A7 P4
     28: MOVER R2 A7 A8
     29: DEJARPERSONA R2 A8 P4
     30: COGERPERSONA R2 A8 P1
     31: DEJARPERSONA R2 A8 P0
     32: MOVER R3 A7 A6
     33: MOVER R2 A8 A3
     34: MOVER R3 A6 A7
     35: DEJARPERSONA R2 A3 P1
     36: MOVER R3 A7 A8
     37: DEJARPERSONA R3 A8 P2
     38: MOVER R3 A8 A7
     39: MOVER R3 A7 A4
     40: DEJARPERSONA R3 A4 P8
     41: MOVER R3 A4 A7
     42: MOVER R3 A7 AL5
     43: COGERSUMINISTRO R3 AL5 S6
     44: MOVER R3 AL5 A7
     45: MOVER R3 A7 A6
     46: MOVER R3 A6 AL2
     47: MOVER R2 A3 A5
     48: MOVER R2 A5 A4
     49: DEJARSUMINISTRO R3 AL2 S6
     50: COGERSUMINISTRO R1 AL3 S5
     51: MOVER R0 AL0 AL4
     52: MOVER R0 AL4 AL3
     53: COGERSUMINISTRO R0 AL3 S1
     54: MOVER R0 AL3 AL1
     55: MOVER R1 AL3 AL1

```

```

56: DEJARSUMINISTRO R1 AL1 S5
57: DEJARSUMINISTRO R0 AL1 S1
58: MOVER R1 AL1 AL3
59: MOVER R1 AL3 AL4
60: COGERSUMINISTRO R1 AL4 S9
61: MOVER R1 AL4 AL3
62: MOVER R1 AL3 AL1
63: DEJARSUMINISTRO R1 AL1 S9
64: MOVER R0 AL1 AL3
65: COGERSUMINISTRO R0 AL3 S7
66: MOVER R0 AL3 AL4
67: MOVER R0 AL4 AL0
68: DEJARSUMINISTRO R0 AL0 S7
69: COGERSUMINISTRO R0 AL0 S8
70: MOVER R0 AL0 AL4
71: MOVER R0 AL4 AL3
72: MOVER R0 AL3 AL1
73: DEJARSUMINISTRO R0 AL1 S8
74: MOVER R2 A4 A7
75: MOVER R2 A7 AL5
76: COGERSUMINISTRO R2 AL5 S2
77: MOVER R2 AL5 A7
78: MOVER R2 A7 A4
79: MOVER R2 A4 A5
80: MOVER R2 A5 AL3
81: MOVER R3 AL2 A6
82: MOVER R3 A6 A7
83: DEJARSUMINISTRO R2 AL3 S2
84: MOVER R3 A7 A4
85: COGERPERSONA R3 A4 P7
86: MOVER R3 A4 A7
87: MOVER R3 A7 AL5
88: MOVER R3 AL5 A0
89: MOVER R3 A0 A1
90: COGERPERSONA R3 A1 P6
91: MOVER R3 A1 A0
92: MOVER R3 A0 A2
93: DEJARPERSONA R3 A2 P7
94: COGERPERSONA R3 A2 P3
95: MOVER R3 A2 A0
96: MOVER R3 A0 AL5
97: MOVER R3 AL5 A7
98: MOVER R3 A7 A4
99: DEJARPERSONA R3 A4 P3
100: MOVER R3 A4 A5
101: DEJARPERSONA R3 A5 P6

```

```

time spent:  0.00 seconds instantiating 2544 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 275 facts and 472 actions
            0.00 seconds creating final representation with 270 relevant facts, 13
relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.17 seconds searching, evaluating 3048 states, to a max depth of 5
            0.17 seconds total time

```

## Resultado

Como se puede observar, las cargas se reparten bien entre los rovers, y en ningún caso se incumple la restricción de que necesitan combustible para moverse. De las 73 unidades de combustible iniciales, han consumido un 62, un 84.9%. Los rovers se han repartido la cantidad de movimientos de manera más o menos equitativa, repartidos de esta manera:

- Rover 0: Movimiento: 17.8%, Combustible gastado: 61.1%
- Rover 1: Movimiento: 16.1%, Combustible gastado: 100%
- Rover 2: Movimiento: 24.2%, Combustible gastado: 78.9%
- Rover 3: Movimiento: 42%, Combustible gastado: 100%

Es notable que el rover 3 realiza más trabajo que los demás, y este hecho se puede explicar debido a que cuanto más se mueva un rover, mientras le quede combustible, más beneficioso es que se siga moviendo, debido a que desde el nuevo punto en el que está puede realizar cosas nuevas. También podemos ver como se intenta apurar al máximo la cantidad de combustible de los rovers, con tal de minimizar el número de pasos (métrica implícita).

## Prueba 6

Este juego de prueba es para la segunda versión de la extensión 2 del proyecto.

### Introducción

Este juego también se ha creado con el generador de juegos de prueba y contiene 20 recursos (6 personas y 14 suministros), 15 bases (6 asentamientos y 9 almacenes), 6 rovers y 29 peticiones. En esta versión de la extensión sí se busca minimizar el combustible gastado, con lo que esperamos ver consumido un porcentaje menor que el test anterior. Al tener el mismo número de recursos y bases que en el test 5, se podrán comparar cómodamente los resultados. Los rovers tienen inicialmente 2, 14, 25, 15, 6 y 7 unidades de combustible respectivamente.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-ext2)
  (:objects
    p0 p1 p2 p3 p4 p5 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 - suministro
    a0 a1 a2 a3 a4 a5 - asentamiento
```

```

    a10 a11 a12 a13 a14 a15 a16 a17 a18 - almacen
    r0 r1 r2 r3 r4 r5 - rover
)
(:init
  (basesAdyacentes a0 a2)
  (basesAdyacentes a0 a5)
  (basesAdyacentes a0 a17)
  (basesAdyacentes a0 a18)
  (basesAdyacentes a1 a16)
  (basesAdyacentes a2 a0)
  (basesAdyacentes a2 a3)
  (basesAdyacentes a2 a4)
  (basesAdyacentes a3 a2)
  (basesAdyacentes a3 a14)
  (basesAdyacentes a3 a4)
  (basesAdyacentes a3 a5)
  (basesAdyacentes a4 a2)
  (basesAdyacentes a4 a3)
  (basesAdyacentes a4 a15)
  (basesAdyacentes a4 a16)
  (basesAdyacentes a4 a18)
  (basesAdyacentes a5 a0)
  (basesAdyacentes a5 a13)
  (basesAdyacentes a5 a3)
  (basesAdyacentes a10 a13)
  (basesAdyacentes a10 a11)
  (basesAdyacentes a11 a15)
  (basesAdyacentes a11 a16)
  (basesAdyacentes a11 a17)
  (basesAdyacentes a11 a10)
  (basesAdyacentes a12 a13)
  (basesAdyacentes a12 a16)
  (basesAdyacentes a13 a12)
  (basesAdyacentes a13 a17)
  (basesAdyacentes a13 a5)
  (basesAdyacentes a13 a10)
  (basesAdyacentes a14 a3)
  (basesAdyacentes a14 a18)
  (basesAdyacentes a15 a4)
  (basesAdyacentes a15 a17)
  (basesAdyacentes a15 a11)
  (basesAdyacentes a16 a12)
  (basesAdyacentes a16 a1)
  (basesAdyacentes a16 a4)
  (basesAdyacentes a16 a11)
  (basesAdyacentes a17 a0)
  (basesAdyacentes a17 a11)
  (basesAdyacentes a17 a13)
  (basesAdyacentes a17 a15)
  (basesAdyacentes a17 a18)
  (basesAdyacentes a18 a0)
  (basesAdyacentes a18 a14)
  (basesAdyacentes a18 a4)
  (basesAdyacentes a18 a17)
  (posicionRecurso p0 a4)
  (posicionRecurso p1 a4)

```



```

(posicionRecurso p2 a2)
(posicionRecurso p3 a0)
(posicionRecurso p4 a3)
(posicionRecurso p5 a0)
(posicionRecurso s0 a17)
(posicionRecurso s1 a18)
(posicionRecurso s2 a14)
(posicionRecurso s3 a16)
(posicionRecurso s4 a12)
(posicionRecurso s5 a17)
(posicionRecurso s6 a13)
(posicionRecurso s7 a15)
(posicionRecurso s8 a12)
(posicionRecurso s9 a14)
(posicionRecurso s10 a14)
(posicionRecurso s11 a10)
(posicionRecurso s12 a13)
(posicionRecurso s13 a16)
(peticion p0 a3)
(peticion p1 a3)
(peticion p2 a2)
(peticion p3 a2)
(peticion p4 a2)
(peticion p5 a2)
(peticion s0 a16)
(peticion s1 a14)
(peticion s2 a14)
(peticion s3 a15)
(peticion s4 a12)
(peticion s5 a13)
(peticion s6 a14)
(peticion s7 a14)
(peticion s8 a14)
(peticion s9 a14)
(peticion s10 a12)
(peticion s11 a17)
(peticion s12 a10)
(peticion s13 a16)
(peticion s8 a16)
(peticion p4 a3)
(peticion p2 a5)
(peticion p3 a2)
(peticion p3 a1)
(peticion s11 a12)
(peticion s13 a14)
(peticion p3 a4)
(peticion s2 a14)
(posicionRover r0 a18) (= (numPersonas r0) 0) (= (combustible r0) 2)
(posicionRover r1 a18) (= (numPersonas r1) 0) (= (combustible r1) 14)
(posicionRover r2 a15) (= (numPersonas r2) 0) (= (combustible r2) 25)
(posicionRover r3 a17) (= (numPersonas r3) 0) (= (combustible r3) 15)
(posicionRover r4 a13) (= (numPersonas r4) 0) (= (combustible r4) 6)
(posicionRover r5 a11) (= (numPersonas r5) 0) (= (combustible r5) 7)
(= (sumaCombustible) 69)
)
(:goal (forall (?r - recurso) (entregado ?r)))

```

```
(:metric maximize (sumaCombustible))  
)
```

## Salida FF

```
ff: parsing domain file  
domain 'DESPLAZAMIENTOS-EXT2' defined  
... done.  
ff: parsing problem file  
problem 'NOMBRE' defined  
... done.  
  
no optimization required. skipping criterion.  
  
no metric specified. plan length assumed.  
  
checking for cyclic := effects --- OK.  
  
ff: search configuration is EHC, if that fails then best-first on  $1 \cdot g(s) + 5 \cdot h(s)$  where  
metric is plan length  
  
Cueing down from goal distance: 63 into depth [1]  
61 [1]  
59 [1]  
58 [1][2]  
57 [1]  
56 [1][2]  
55 [1]  
54 [1][2][3]  
53 [1]  
52 [1]  
51 [1][2][3]  
50 [1]  
49 [1][2]  
48 [1]  
47 [1]  
46 [1][2]  
45 [1]  
44 [1]  
42 [1]  
41 [1]  
40 [1]  
39 [1]  
38 [1][2]  
37 [1]  
36 [1]  
35 [1]  
34 [1][2]  
33 [1]  
32 [1]  
31 [1]  
30 [1]
```

|    |                 |
|----|-----------------|
| 29 | [1]             |
| 28 | [1][2][3]       |
| 26 | [1]             |
| 25 | [1][2][3]       |
| 24 | [1]             |
| 23 | [1][2]          |
| 22 | [1][2]          |
| 21 | [1][2]          |
| 20 | [1][2][3][4]    |
| 19 | [1]             |
| 18 | [1]             |
| 17 | [1]             |
| 16 | [1]             |
| 15 | [1][2]          |
| 14 | [1][2][3][4]    |
| 13 | [1]             |
| 12 | [1][2]          |
| 11 | [1][2][3][4][5] |
| 9  | [1][2][3]       |
| 8  | [1][2][3][4][5] |
| 7  | [1][2]          |
| 6  | [1]             |
| 5  | [1]             |
| 4  | [1]             |
| 3  | [1]             |
| 2  | [1]             |
| 1  | [1]             |
| 0  |                 |

ff: found legal plan as follows

```

step  0: MOVER R3 AL7 AL8
      1: MOVER R4 AL3 AL7
      2: MOVER R4 AL7 AL1
      3: MOVER R4 AL1 AL6
      4: COGERSUMINISTRO R4 AL6 S3
      5: MOVER R5 AL1 AL0
      6: MOVER R5 AL0 AL3
      7: MOVER R5 AL3 AL2
      8: COGERSUMINISTRO R5 AL2 S8
      9: MOVER R5 AL2 AL6
     10: DEJARSUMINISTRO R5 AL6 S8
     11: MOVER R5 AL6 AL2
     12: COGERSUMINISTRO R5 AL2 S4
     13: DEJARSUMINISTRO R5 AL2 S4
     14: MOVER R5 AL2 AL3
     15: COGERSUMINISTRO R5 AL3 S12
     16: MOVER R5 AL3 AL0
     17: DEJARSUMINISTRO R5 AL0 S12
     18: MOVER R2 AL5 AL1
     19: MOVER R2 AL1 AL0
     20: COGERSUMINISTRO R2 AL0 S11
     21: MOVER R2 AL0 AL3
     22: MOVER R3 AL8 A0
     23: COGERPERSONA R3 A0 P5
     24: COGERPERSONA R3 A0 P3

```

25: MOVER R3 A0 A2  
26: DEJARPERSONA R3 A2 P5  
27: COGERPERSONA R3 A2 P2  
28: DEJARPERSONA R3 A2 P3  
29: DEJARPERSONA R3 A2 P2  
30: MOVER R2 AL3 AL2  
31: DEJARSUMINISTRO R2 AL2 S11  
32: MOVER R2 AL2 AL6  
33: COGERSUMINISTRO R2 AL6 S13  
34: DEJARSUMINISTRO R2 AL6 S13  
35: COGERSUMINISTRO R1 AL8 S1  
36: MOVER R1 AL8 AL4  
37: DEJARSUMINISTRO R1 AL4 S1  
38: COGERSUMINISTRO R1 AL4 S9  
39: DEJARSUMINISTRO R1 AL4 S9  
40: COGERSUMINISTRO R1 AL4 S2  
41: DEJARSUMINISTRO R1 AL4 S2  
42: MOVER R4 AL6 A4  
43: MOVER R0 AL8 AL7  
44: MOVER R4 A4 AL5  
45: DEJARSUMINISTRO R4 AL5 S3  
46: MOVER R3 A2 A3  
47: COGERPERSONA R3 A3 P4  
48: DEJARPERSONA R3 A3 P4  
49: MOVER R3 A3 AL4  
50: COGERSUMINISTRO R0 AL7 S5  
51: MOVER R4 AL5 AL7  
52: MOVER R0 AL7 AL3  
53: DEJARSUMINISTRO R0 AL3 S5  
54: MOVER R2 AL6 A4  
55: MOVER R2 A4 AL8  
56: MOVER R2 AL8 AL7  
57: MOVER R3 AL4 AL8  
58: MOVER R3 AL8 A4  
59: COGERPERSONA R3 A4 P1  
60: COGERPERSONA R3 A4 P0  
61: MOVER R3 A4 A3  
62: DEJARPERSONA R3 A3 P1  
63: DEJARPERSONA R3 A3 P0  
64: MOVER R2 AL7 AL3  
65: MOVER R3 A3 AL4  
66: MOVER R2 AL3 AL7  
67: COGERSUMINISTRO R2 AL7 S0  
68: MOVER R2 AL7 AL1  
69: MOVER R2 AL1 AL6  
70: DEJARSUMINISTRO R2 AL6 S0  
71: MOVER R2 AL6 A4  
72: MOVER R2 A4 AL8  
73: MOVER R2 AL8 AL7  
74: MOVER R2 AL7 AL5  
75: COGERSUMINISTRO R2 AL5 S7  
76: MOVER R2 AL5 AL7  
77: MOVER R2 AL7 AL8  
78: MOVER R2 AL8 AL4  
79: DEJARSUMINISTRO R2 AL4 S7  
80: MOVER R2 AL4 A3

```

81: MOVER R2 A3 A5
82: MOVER R2 A5 AL3
83: COGERSUMINISTRO R2 AL3 S6
84: MOVER R2 AL3 AL7
85: MOVER R2 AL7 AL8
86: MOVER R2 AL8 AL4
87: COGERSUMINISTRO R3 AL4 S10
88: MOVER R3 AL4 AL8
89: MOVER R3 AL8 A4
90: MOVER R3 A4 AL6
91: MOVER R3 AL6 AL2
92: DEJARSUMINISTRO R3 AL2 S10
93: DEJARSUMINISTRO R2 AL4 S6

```

```

time spent:  0.00 seconds instantiating 3900 easy, 0 hard action templates
            0.00 seconds reachability analysis, yielding 332 facts and 708 actions
            0.00 seconds creating final representation with 330 relevant facts, 19
relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.13 seconds searching, evaluating 2019 states, to a max depth of 5
            0.13 seconds total time

```

## Resultado

Como se puede observar, se siguen respetando las restricciones de combustible, y esta vez se han consumido 54 de 69 unidades de combustible, un 78.2%. Este valor es menor que el del test 5, validando que se busca minimizar el combustible gastado. Esta vez, los rovers se han repartido la cantidad de movimiento de esta manera:

- Rover 0: Movimiento 1.8%, Combustible gastado: 100%
- Rover 1: Movimiento 3.7%, Combustible gastado: 7.1%
- Rover 2: Movimiento 46.3%, Combustible gastado: 100%
- Rover 3: Movimiento 24.1%, Combustible gastado: 86.7%
- Rover 4: Movimiento 11.1%, Combustible gastado: 100%
- Rover 5: Movimiento 13%, Combustible gastado: 100%

En este juego de pruebas es aún más obvio que el movimiento favorece el movimiento, con el rover 2 cubriendo casi la mitad de estos mientras que los rovers 0 y 1 prácticamente no se han movido. También es cierto que condiciones iniciales generadas aleatoriamente han dado una cantidad de rovers que quizás es excesiva para resolver el plan actual. La cantidad de combustible gastado de cada rover nos deja ver que estos valores están mucho más polarizados, comparado con los del juego de pruebas 5. Es decir, los rovers buscan gastar toda o nada de su combustible. Esto se puede explicar por la reducción de movimientos menos útiles al intentar conservar combustible, y valida los resultados del test.

## Prueba 7

Este juego de prueba es para la primera versión de la extensión 3 del proyecto.

### Introducción

En esta prueba (generada de nuevo con el programa generador de juegos de prueba) se cuenta con 8 personas, 14 suministros, 5 asentamientos, 8 almacenes, 5 rovers y 30 peticiones. Esta prueba está generada para la extensión 3 del proyecto, es decir, incluye un sistema de prioridades de cada petición, por lo que deberíamos notar como los rovers a veces hacen trayectos más largos para entregar los recursos en sitios donde cumplan pedidos de mayor prioridad.

### Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-ext3)
  (:objects
    p0 p1 p2 p3 p4 p5 p6 p7 - persona
    s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s13 - suministro
    a0 a1 a2 a3 a4 - asentamiento
    al0 al1 al2 al3 al4 al5 al6 al7 - almacen
    r0 r1 r2 r3 r4 - rover
  )
  (:init
    (basesAdyacentes a0 a2)
    (basesAdyacentes a0 a3)
    (basesAdyacentes a0 a4)
    (basesAdyacentes a0 al1)
    (basesAdyacentes a0 al3)
    (basesAdyacentes a1 a3)
    (basesAdyacentes a1 al0)
    (basesAdyacentes a1 al3)
    (basesAdyacentes a1 al5)
    (basesAdyacentes a1 al6)
    (basesAdyacentes a1 al7)
    (basesAdyacentes a2 a0)
    (basesAdyacentes a2 al0)
    (basesAdyacentes a2 al3)
    (basesAdyacentes a2 al4)
    (basesAdyacentes a2 al5)
    (basesAdyacentes a2 al6)
    (basesAdyacentes a3 a0)
    (basesAdyacentes a3 a1)
    (basesAdyacentes a3 a4)
    (basesAdyacentes a3 al1)
    (basesAdyacentes a3 al5)
    (basesAdyacentes a4 a0)
    (basesAdyacentes a4 a3)
    (basesAdyacentes a4 al0)
```

```
(basesAdyacentes a4 a11)
(basesAdyacentes a4 a14)
(basesAdyacentes a4 a15)
(basesAdyacentes a4 a16)
(basesAdyacentes a10 a1)
(basesAdyacentes a10 a2)
(basesAdyacentes a10 a4)
(basesAdyacentes a10 a13)
(basesAdyacentes a10 a17)
(basesAdyacentes a11 a0)
(basesAdyacentes a11 a3)
(basesAdyacentes a11 a4)
(basesAdyacentes a11 a14)
(basesAdyacentes a11 a15)
(basesAdyacentes a11 a17)
(basesAdyacentes a12 a14)
(basesAdyacentes a13 a0)
(basesAdyacentes a13 a1)
(basesAdyacentes a13 a2)
(basesAdyacentes a13 a10)
(basesAdyacentes a13 a14)
(basesAdyacentes a13 a17)
(basesAdyacentes a14 a2)
(basesAdyacentes a14 a4)
(basesAdyacentes a14 a11)
(basesAdyacentes a14 a12)
(basesAdyacentes a14 a13)
(basesAdyacentes a14 a15)
(basesAdyacentes a14 a16)
(basesAdyacentes a14 a17)
(basesAdyacentes a15 a1)
(basesAdyacentes a15 a2)
(basesAdyacentes a15 a3)
(basesAdyacentes a15 a4)
(basesAdyacentes a15 a11)
(basesAdyacentes a15 a14)
(basesAdyacentes a15 a17)
(basesAdyacentes a16 a1)
(basesAdyacentes a16 a2)
(basesAdyacentes a16 a4)
(basesAdyacentes a16 a14)
(basesAdyacentes a17 a1)
(basesAdyacentes a17 a10)
(basesAdyacentes a17 a11)
(basesAdyacentes a17 a13)
(basesAdyacentes a17 a14)
(basesAdyacentes a17 a15)
(posicionRecurso p0 a2)
(posicionRecurso p1 a1)
(posicionRecurso p2 a4)
(posicionRecurso p3 a4)
(posicionRecurso p4 a1)
(posicionRecurso p5 a0)
(posicionRecurso p6 a2)
(posicionRecurso p7 a4)
(posicionRecurso s0 a13)
```

```

(posicionRecurso s1 a17)
(posicionRecurso s2 a11)
(posicionRecurso s3 a12)
(posicionRecurso s4 a10)
(posicionRecurso s5 a14)
(posicionRecurso s6 a11)
(posicionRecurso s7 a14)
(posicionRecurso s8 a15)
(posicionRecurso s9 a12)
(posicionRecurso s10 a14)
(posicionRecurso s11 a16)
(posicionRecurso s12 a15)
(posicionRecurso s13 a12)
(peticion p0 a4) (= (prioridad p0 a4) 3)
(peticion p1 a3) (= (prioridad p1 a3) 1)
(peticion p2 a2) (= (prioridad p2 a2) 3)
(peticion p3 a2) (= (prioridad p3 a2) 2)
(peticion p4 a4) (= (prioridad p4 a4) 1)
(peticion p5 a3) (= (prioridad p5 a3) 1)
(peticion p6 a0) (= (prioridad p6 a0) 3)
(peticion p7 a1) (= (prioridad p7 a1) 1)
(peticion s0 a12) (= (prioridad s0 a12) 1)
(peticion s1 a12) (= (prioridad s1 a12) 3)
(peticion s2 a14) (= (prioridad s2 a14) 3)
(peticion s3 a17) (= (prioridad s3 a17) 2)
(peticion s4 a16) (= (prioridad s4 a16) 1)
(peticion s5 a13) (= (prioridad s5 a13) 3)
(peticion s6 a14) (= (prioridad s6 a14) 3)
(peticion s7 a16) (= (prioridad s7 a16) 2)
(peticion s8 a11) (= (prioridad s8 a11) 3)
(peticion s9 a12) (= (prioridad s9 a12) 3)
(peticion s10 a11) (= (prioridad s10 a11) 3)
(peticion s11 a12) (= (prioridad s11 a12) 3)
(peticion s12 a13) (= (prioridad s12 a13) 1)
(peticion s13 a14) (= (prioridad s13 a14) 2)
(peticion p3 a4) (= (prioridad p3 a4) 1)
(peticion s13 a17) (= (prioridad s13 a17) 1)
(peticion s13 a14) (= (prioridad s13 a14) 1)
(peticion s13 a13) (= (prioridad s13 a13) 1)
(peticion p7 a4) (= (prioridad p7 a4) 2)
(peticion p2 a0) (= (prioridad p2 a0) 3)
(peticion p2 a3) (= (prioridad p2 a3) 2)
(peticion p2 a0) (= (prioridad p2 a0) 2)
(posicionRover r0 a11) (= (numPersonas r0) 0) (= (combustible r0) 8)
(posicionRover r1 a10) (= (numPersonas r1) 0) (= (combustible r1) 9)
(posicionRover r2 a0) (= (numPersonas r2) 0) (= (combustible r2) 21)
(posicionRover r3 a15) (= (numPersonas r3) 0) (= (combustible r3) 23)
(posicionRover r4 a16) (= (numPersonas r4) 0) (= (combustible r4) 24)
(= (sumaCombustible) 85)
(= (sumaPrioridad) 0)
)
(:goal (forall (?r - recurso) (entregado ?r)))
(:metric maximize (sumaPrioridad))
)

```



## Salida FF

```
ff: parsing domain file
domain 'DESPLAZAMIENTOS-EXT3' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

no optimization required. skipping criterion.

no metric specified. plan length assumed.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on  $1*g(s) + 5*h(s)$  where
metric is plan length

Cueing down from goal distance: 68 into depth [1]
62 [1]
61 [1]
60 [1]
59 [1]
58 [1][2]
56 [1]
55 [1][2]
52 [1]
51 [1][2][3]
50 [1]
49 [1][2][3]
48 [1][2]
47 [1]
46 [1]
45 [1][2]
44 [1]
43 [1][2]
42 [1]
41 [1]
40 [1]
39 [1][2]
38 [1][2]
37 [1]
36 [1]
35 [1][2][3]
34 [1][2]
33 [1]
32 [1]
31 [1][2][3]
30 [1]
29 [1]
28 [1]
27 [1]
26 [1]
```

|    |              |
|----|--------------|
| 25 | [1]          |
| 24 | [1]          |
| 23 | [1]          |
| 22 | [1]          |
| 21 | [1]          |
| 20 | [1]          |
| 19 | [1][2][3]    |
| 18 | [1][2]       |
| 17 | [1][2][3]    |
| 16 | [1]          |
| 15 | [1][2][3][4] |
| 14 | [1]          |
| 13 | [1][2]       |
| 12 | [1][2][3][4] |
| 11 | [1]          |
| 10 | [1]          |
| 9  | [1][2]       |
| 8  | [1]          |
| 7  | [1]          |
| 6  | [1][2][3]    |
| 5  | [1]          |
| 4  | [1]          |
| 3  | [1]          |
| 2  | [1]          |
| 1  | [1]          |
| 0  |              |

ff: found legal plan as follows

```

step  0: MOVER R3 AL5 AL7
      1: COGERSUMINISTRO R0 AL1 S6
      2: COGERPERSONA R2 A0 P5
      3: COGERSUMINISTRO R1 AL0 S4
      4: MOVER R1 AL0 A1
      5: MOVER R4 AL6 AL4
      6: COGERSUMINISTRO R4 AL4 S7
      7: MOVER R3 AL7 AL4
      8: COGERSUMINISTRO R3 AL4 S10
      9: MOVER R0 AL1 AL4
     10: DEJARSUMINISTRO R0 AL4 S6
     11: MOVER R2 A0 A2
     12: COGERPERSONA R2 A2 P6
     13: MOVER R2 A2 A0
     14: DEJARPERSONA R2 A0 P6
     15: MOVER R2 A0 A2
     16: MOVER R2 A2 AL5
     17: MOVER R1 A1 AL6
     18: MOVER R2 AL5 A1
     19: COGERPERSONA R2 A1 P1
     20: MOVER R4 AL4 AL6
     21: DEJARSUMINISTRO R4 AL6 S7
     22: MOVER R3 AL4 AL1
     23: DEJARSUMINISTRO R3 AL1 S10
     24: COGERSUMINISTRO R3 AL1 S2
     25: MOVER R3 AL1 AL4
     26: DEJARSUMINISTRO R3 AL4 S2

```

27: COGERSUMINISTRO R3 AL4 S5  
28: MOVER R3 AL4 AL3  
29: DEJARSUMINISTRO R3 AL3 S5  
30: DEJARSUMINISTRO R1 AL6 S4  
31: MOVER R1 AL6 AL4  
32: MOVER R1 AL4 AL5  
33: COGERSUMINISTRO R1 AL5 S8  
34: MOVER R1 AL5 AL1  
35: DEJARSUMINISTRO R1 AL1 S8  
36: MOVER R4 AL6 A1  
37: COGERPERSONA R4 A1 P4  
38: MOVER R4 A1 AL5  
39: MOVER R2 A1 A3  
40: DEJARPERSONA R2 A3 P5  
41: MOVER R2 A3 A4  
42: COGERPERSONA R2 A4 P2  
43: MOVER R4 AL5 A2  
44: MOVER R1 AL1 A4  
45: COGERPERSONA R1 A4 P7  
46: COGERPERSONA R1 A4 P3  
47: MOVER R2 A4 A3  
48: COGERPERSONA R4 A2 P0  
49: MOVER R4 A2 AL5  
50: MOVER R4 AL5 A4  
51: DEJARPERSONA R1 A4 P7  
52: DEJARPERSONA R4 A4 P4  
53: DEJARPERSONA R1 A4 P3  
54: DEJARPERSONA R2 A3 P2  
55: DEJARPERSONA R2 A3 P1  
56: DEJARPERSONA R4 A4 P0  
57: MOVER R0 AL4 AL2  
58: MOVER R1 A4 AL4  
59: COGERSUMINISTRO R0 AL2 S9  
60: DEJARSUMINISTRO R0 AL2 S9  
61: MOVER R0 AL2 AL4  
62: MOVER R1 AL4 AL2  
63: COGERSUMINISTRO R1 AL2 S13  
64: MOVER R1 AL2 AL4  
65: DEJARSUMINISTRO R1 AL4 S13  
66: MOVER R0 AL4 AL5  
67: COGERSUMINISTRO R0 AL5 S12  
68: MOVER R0 AL5 A2  
69: MOVER R4 A4 AL4  
70: MOVER R0 A2 AL3  
71: DEJARSUMINISTRO R0 AL3 S12  
72: MOVER R0 AL3 AL4  
73: MOVER R0 AL4 AL6  
74: MOVER R4 AL4 AL6  
75: COGERSUMINISTRO R3 AL3 S0  
76: MOVER R3 AL3 AL4  
77: COGERSUMINISTRO R4 AL6 S11  
78: MOVER R4 AL6 AL4  
79: MOVER R3 AL4 AL2  
80: MOVER R4 AL4 AL2  
81: DEJARSUMINISTRO R4 AL2 S11  
82: DEJARSUMINISTRO R3 AL2 S0

```
83: COGERSUMINISTRO R4 AL2 S3
84: MOVER R4 AL2 AL4
85: MOVER R4 AL4 AL7
86: DEJARSUMINISTRO R4 AL7 S3
87: COGERSUMINISTRO R4 AL7 S1
88: MOVER R4 AL7 AL4
89: MOVER R4 AL4 AL2
90: DEJARSUMINISTRO R4 AL2 S1
```

```
time spent:    0.00 seconds instantiating 3220 easy, 0 hard action templates
              0.00 seconds reachability analysis, yielding 306 facts and 735 actions
              0.00 seconds creating final representation with 304 relevant facts, 17
relevant fluents
              0.00 seconds computing LNF
              0.00 seconds building connectivity graph
              0.23 seconds searching, evaluating 3359 states, to a max depth of 4
              0.23 seconds total time
```

## Resultado

Podemos ver como el programa ha generado un plan con muchas acciones para maximizar todo lo posible la ganancia de prioridades. Si miramos con detalle los pedidos que se entregan, se puede ver que prácticamente todos son de prioridad 2 o 3. Además, notamos como se han usado de forma bastante equitativa todos los rovers, ya que se ha intentado aprovechar todo el combustible sin importar el gasto para realizar los mejores pedidos. El planner ha conseguido un resultado en muy poco tiempo, en parte gracias a la gran cantidad de conexiones entre bases del juego de pruebas.

## Prueba 8

Este juego de prueba es para la segunda versión de la extensión 3 del proyecto.

## Introducción

En este juego de pruebas, se añade a la métrica la suma de combustibles junto con un factor multiplicador, para poder regular si queremos priorizar más que se gaste poca gasolina o que se reparten los pedidos con mayor prioridad. Para este juego de pruebas hemos dado un valor de 10 para la suma de prioridades y un valor de 8 para la suma de gasolina sin gastar. El juego cuenta con 5 personas, 5 suministros, 9 asentamientos, 5 almacenes, 6 rovers y 27 pedidos. El resultado debería ser parecido al del juego de pruebas

anterior, pero añadiendo más movimientos porque el factor de la gasolina vuelve a tener un peso importante.

## Entrada FF

```
(define (problem nombre)
  (:domain desplazamientos-ext3)
  (:objects
    p0 p1 p2 p3 p4 - persona
    s0 s1 s2 s3 s4 - suministro
    a0 a1 a2 a3 a4 a5 a6 a7 a8 - asentamiento
    al0 al1 al2 al3 al4 - almacen
    r0 r1 r2 r3 r4 r5 - rover
  )
  (:init
    (basesAdyacentes a0 al0)
    (basesAdyacentes a1 al4)
    (basesAdyacentes a2 a8)
    (basesAdyacentes a2 a6)
    (basesAdyacentes a3 a8)
    (basesAdyacentes a3 al1)
    (basesAdyacentes a3 al3)
    (basesAdyacentes a3 a6)
    (basesAdyacentes a4 a8)
    (basesAdyacentes a4 al0)
    (basesAdyacentes a5 a8)
    (basesAdyacentes a6 a8)
    (basesAdyacentes a6 a2)
    (basesAdyacentes a6 a3)
    (basesAdyacentes a7 al1)
    (basesAdyacentes a7 al4)
    (basesAdyacentes a8 a2)
    (basesAdyacentes a8 a3)
    (basesAdyacentes a8 a4)
    (basesAdyacentes a8 a5)
    (basesAdyacentes a8 a6)
    (basesAdyacentes al0 a0)
    (basesAdyacentes al0 a4)
    (basesAdyacentes al1 a3)
    (basesAdyacentes al1 al3)
    (basesAdyacentes al1 a7)
    (basesAdyacentes al2 al4)
    (basesAdyacentes al3 al1)
    (basesAdyacentes al3 a3)
    (basesAdyacentes al4 a1)
    (basesAdyacentes al4 al2)
    (basesAdyacentes al4 a7)
    (posicionRecurso p0 a1)
    (posicionRecurso p1 a5)
    (posicionRecurso p2 a1)
    (posicionRecurso p3 a3)
    (posicionRecurso p4 a1)
    (posicionRecurso s0 al1)
    (posicionRecurso s1 al0)
```

```

(posicionRecurso s2 a13)
(posicionRecurso s3 a12)
(posicionRecurso s4 a10)
(peticion p0 a6) (= (prioridad p0 a6) 1)
(peticion p1 a4) (= (prioridad p1 a4) 1)
(peticion p2 a8) (= (prioridad p2 a8) 1)
(peticion p3 a8) (= (prioridad p3 a8) 1)
(peticion p4 a5) (= (prioridad p4 a5) 1)
(peticion s0 a11) (= (prioridad s0 a11) 3)
(peticion s1 a14) (= (prioridad s1 a14) 3)
(peticion s2 a14) (= (prioridad s2 a14) 1)
(peticion s3 a14) (= (prioridad s3 a14) 1)
(peticion s4 a13) (= (prioridad s4 a13) 3)
(peticion s1 a11) (= (prioridad s1 a11) 1)
(peticion p4 a6) (= (prioridad p4 a6) 3)
(peticion s2 a13) (= (prioridad s2 a13) 3)
(peticion p3 a1) (= (prioridad p3 a1) 2)
(peticion s3 a12) (= (prioridad s3 a12) 1)
(peticion p1 a1) (= (prioridad p1 a1) 2)
(peticion s4 a14) (= (prioridad s4 a14) 3)
(peticion p0 a3) (= (prioridad p0 a3) 1)
(peticion s3 a13) (= (prioridad s3 a13) 1)
(peticion s0 a10) (= (prioridad s0 a10) 2)
(peticion p1 a8) (= (prioridad p1 a8) 2)
(peticion s1 a13) (= (prioridad s1 a13) 1)
(peticion s2 a13) (= (prioridad s2 a13) 2)
(peticion p0 a2) (= (prioridad p0 a2) 1)
(peticion s0 a11) (= (prioridad s0 a11) 3)
(peticion s2 a10) (= (prioridad s2 a10) 1)
(peticion p3 a7) (= (prioridad p3 a7) 2)
(posicionRover r0 a4) (= (numPersonas r0) 0) (= (combustible r0) 28)
(posicionRover r1 a2) (= (numPersonas r1) 0) (= (combustible r1) 24)
(posicionRover r2 a2) (= (numPersonas r2) 0) (= (combustible r2) 5)
(posicionRover r3 a11) (= (numPersonas r3) 0) (= (combustible r3) 18)
(posicionRover r4 a7) (= (numPersonas r4) 0) (= (combustible r4) 20)
(posicionRover r5 a0) (= (numPersonas r5) 0) (= (combustible r5) 6)
(= (sumaCombustible) 101)
(= (sumaPrioridad) 0)
)
(:goal (forall (?r - recurso) (entregado ?r)))
(:metric maximize (+ (* 10 (sumaPrioridad)) (* 8 (sumaCombustible))))
)

```

## Salida FF

```

ff: parsing domain file
domain 'DESPLAZAMIENTOS-EXT3' defined
... done.
ff: parsing problem file
problem 'NOMBRE' defined
... done.

```

no optimization required. skipping criterion.

no metric specified. plan length assumed.

checking for cyclic := effects --- OK.

ff: search configuration is EHC, if that fails then best-first on  $1*g(s) + 5*h(s)$  where  
metric is plan length

Cueing down from goal distance: 36 into depth [1]

|    |                    |
|----|--------------------|
| 34 | [1]                |
| 33 | [1]                |
| 32 | [1]                |
| 31 | [1]                |
| 30 | [1]                |
| 29 | [1]                |
| 28 | [1][2][3]          |
| 27 | [1]                |
| 26 | [1]                |
| 25 | [1][2][3]          |
| 24 | [1][2]             |
| 23 | [1][2][3]          |
| 22 | [1][2][3]          |
| 21 | [1][2][3]          |
| 20 | [1]                |
| 19 | [1][2][3][4]       |
| 18 | [1][2]             |
| 17 | [1]                |
| 16 | [1]                |
| 15 | [1]                |
| 14 | [1][2]             |
| 12 | [1][2]             |
| 11 | [1][2]             |
| 10 | [1]                |
| 9  | [1]                |
| 8  | [1][2][3][4][5][6] |
| 7  | [1][2]             |
| 6  | [1]                |
| 5  | [1]                |
| 4  | [1]                |
| 3  | [1]                |
| 2  | [1]                |
| 1  | [1]                |
| 0  |                    |

ff: found legal plan as follows

step 0: MOVER R0 A4 A8  
1: COGERSUMINISTRO R3 AL1 S0  
2: DEJARSUMINISTRO R3 AL1 S0  
3: MOVER R3 AL1 AL3  
4: COGERSUMINISTRO R3 AL3 S2  
5: MOVER R4 A7 AL4

6: DEJARSUMINISTRO R3 AL3 S2  
 7: MOVER R0 A8 A4  
 8: MOVER R4 AL4 AL2  
 9: MOVER R3 AL3 AL1  
 10: COGERSUMINISTRO R4 AL2 S3  
 11: DEJARSUMINISTRO R4 AL2 S3  
 12: MOVER R3 AL1 A3  
 13: COGERPERSONA R3 A3 P3  
 14: MOVER R3 A3 AL1  
 15: MOVER R3 AL1 A7  
 16: DEJARPERSONA R3 A7 P3  
 17: MOVER R3 A7 AL4  
 18: MOVER R3 AL4 A1  
 19: COGERPERSONA R3 A1 P0  
 20: COGERPERSONA R3 A1 P4  
 21: MOVER R3 A1 AL4  
 22: MOVER R3 AL4 A7  
 23: MOVER R3 A7 AL1  
 24: MOVER R0 A4 AL0  
 25: MOVER R3 AL1 A3  
 26: DEJARPERSONA R3 A3 P0  
 27: MOVER R3 A3 A6  
 28: MOVER R0 AL0 A4  
 29: MOVER R4 AL2 AL4  
 30: DEJARPERSONA R3 A6 P4  
 31: MOVER R4 AL4 A1  
 32: COGERPERSONA R4 A1 P2  
 33: MOVER R4 A1 AL4  
 34: MOVER R4 AL4 A7  
 35: MOVER R4 A7 AL1  
 36: MOVER R4 AL1 A3  
 37: MOVER R0 A4 AL0  
 38: MOVER R4 A3 A8  
 39: DEJARPERSONA R4 A8 P2  
 40: MOVER R4 A8 A5  
 41: COGERPERSONA R4 A5 P1  
 42: MOVER R4 A5 A8  
 43: DEJARPERSONA R4 A8 P1  
 44: MOVER R4 A8 A4  
 45: MOVER R4 A4 AL0  
 46: COGERSUMINISTRO R4 AL0 S4  
 47: MOVER R4 AL0 A4  
 48: MOVER R4 A4 A8  
 49: MOVER R4 A8 A3  
 50: MOVER R4 A3 AL3  
 51: COGERSUMINISTRO R0 AL0 S1  
 52: MOVER R0 AL0 A4  
 53: MOVER R0 A4 A8  
 54: MOVER R0 A8 A3  
 55: MOVER R0 A3 AL3  
 56: DEJARSUMINISTRO R4 AL3 S4  
 57: DEJARSUMINISTRO R0 AL3 S1

time spent: 0.00 seconds instantiating 1872 easy, 0 hard action templates  
 0.00 seconds reachability analysis, yielding 235 facts and 534 actions



```
0.00 seconds creating final representation with 233 relevant facts, 20
relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.02 seconds searching, evaluating 648 states, to a max depth of 6
0.02 seconds total time
```

## Resultado

El resultado muestra un plan con pocos movimientos (ya que había pocos recursos), y si analizamos los pedidos que se han entregado mayoritariamente son los que tienen más prioridad, pero menos que en el juego anterior, pues muchas veces se ha preferido entregar un mismo paquete en una base más cercana aunque la recompensa sea menor.

## Generador de Juegos de Prueba

### Introducción

El script que genera los juegos de prueba lo hemos programado en Python. Este te permite generar juegos de prueba válidos tanto para el nivel básico como para las tres extensiones, incluyendo ambas versiones de las extensiones 2 y 3. El programa se divide en 2 partes:

- Recopilación de datos para personalizar el juego de pruebas.
- Generación de la estructura del problema (El archivo se va creando a medida que esta parte va teniendo las distintas partes del juego de pruebas listas).

## Algoritmo

### Recopilación de datos

En primer lugar, el programa pregunta si los datos se quieren generar de forma aleatoria. En caso afirmativo, pide para que extensión se quiere hacer el juego de pruebas y genera los siguientes datos de forma aleatoria dentro de los siguientes rangos:

- La cantidad de rovers [2, 7]
- La cantidad de peticiones [5, 19]
- La cantidad de personal disponible [5, 14]
- La cantidad de suministros disponibles [4, 14]
- La cantidad de asentamientos [5, 9]
- La cantidad de almacenes [5, 9]

En caso de no querer generar los datos de forma aleatoria, pide todos los datos anteriores al usuario.

### Generación de la estructura del problema

En primer lugar, se declaran todos los objetos según los datos obtenidos de la sección anterior: *persona*, *suministro*, *asentamiento*, *almacen* y *rover*.

Seguidamente, se genera el grafo de relaciones entre las distintas bases (asentamientos y almacenes). El enunciado nos pide que el grafo sea conexo, pero que no esté completamente conectado. Además, dice que no necesita tener muchas conexiones. Debido a estas especificaciones que se nos dan, hemos determinado que la mejor opción era crear un grafo que añada aristas entre dos bases de forma aleatoria hasta tener un grafo conexo. De esta forma, conseguimos un grafo aleatorio, conexo y que tiende a no tener muchas aristas.

Hemos decidido dividir esta parte en tres fases para optimizar el código:

- 1a fase: Crear un grafo con todas las bases, pero sin ninguna conexión entre ellas.
- 2a fase: Añadir tantas aristas como el número de bases - 1. Esto lo hacemos porque para que un grafo sea conexo, es necesario un mínimo de  $n-1$  aristas, siendo  $n$  la cantidad de vértices de este.
- 3a fase: Añadir aristas de forma aleatoria hasta tener un grafo conexo.

En un principio, teníamos pensado no hacer la fase 2, ya que el objetivo del grafo es que sea conexo (Por lo que con la fase 1 y 3 hay suficiente). Sin embargo, debido a que sabemos que el número de aristas mínimo es  $n-1$  y que comprobar que un grafo es conexo tiene un coste relativamente alto, sobre todo cuando hay muchas bases, hemos decidido que era importante implementar, además, la fase 2.

```
#nBases = numero de bases
grafo = nodos_sin_aristas # Fase1

# Se genera un grafo con nBases - 1, ya que es el mínimo
#número de aristas que necesita un grafo para ser conexo
# Fase 2
while grafo.numero_aristas() < nBases-1:
    grafo.setNuevaAristaAleatoria()

# Se generan aristas aleatorias hasta que tengamos un
# Fase3
#grafo conexo
while not grafo.conex():
    grafo.setNuevaAristaAleatoria()
```

Después de generar todas las conexiones entre las distintas bases, se posiciona todo el personal y a todos los suministros de forma aleatoria. Además, también se genera una petición para cada unidad de personal y para cada suministro. Esto último se hace para asegurarnos de que todos los recursos tienen al menos una petición, ya que suponemos que si ha llegado a marte es porque es necesario.

```
nPersonal = número de unidades de personal disponible
nSuministros = número de suministros disponibles
Por cada número de 0 a nPersonal:
    Elegimos un asentamiento de forma aleatoria
    Se crea una petición para este personal de forma aleatoria

Por cada número de 0 a nSuministros:
    Elegimos un almacén de forma aleatoria
    Se crea una petición para este suministro de forma aleatoria
```

Después de asignar cada recurso a una base, se generan las distintas peticiones.

```
nPeticiones = número extra de peticiones
Por cada número de 0 a nPeticiones:
    Se escoge un recurso al azar
    Si recurso_escogido = personal:
        Se escoge un asentamiento al azar
    Si recurso_escogido = suministro:
        Se escoge un almacén a azar
    Se genera una petición del recurso escogido a la base escogida
```

De la misma forma que con los recursos, se posiciona a cada rover en una base escogida aleatoriamente:

```
nRovers = número total de bases
nBases = número total de bases
Por cada número de 0 a nRovers:
    Elegimos una base de forma aleatoria
    si extension >= 1:
        numPersonas = 0
    si extension >= 2:
        combustible = número aleatorio entre 0 y nBases*2+1
```

Por último, dependiendo de la extensión, se declaran las distintas funciones y distintas métricas:

```
si extension >= 2:
```

```
    sumaCombustible = la suma del combustible de todos los rovers
Si extension >= 3:
    sumaPrioridad = 0
Si extension = 2 ampliada:
    Se pide maximizar el combustible restante
Si extension = 3 básica:
    Se pide maximizar la prioridad servida
Si extension = 3 ampliada:
    Se pide maximizar la prioridad servida y el combustible restante
```

# Estudio de la evolución del tiempo de resolución

## Prueba 1

En este caso, hemos decidido estudiar la variación en el tiempo de resolución del programa al incrementar la cantidad de rovers. Hemos definido los parámetros de los juegos de prueba de forma que el tiempo de ejecución se vea alterado mayormente por la variación en la cantidad de rovers. Hemos decidido establecer una cantidad de recursos, personal y bases igual en todo el experimento y dejar que la posición de estos, así como las peticiones se elijan de forma aleatoria. El criterio seguido para elegir la cantidad de cada elemento ha sido el mencionado por el enunciado (*“La vida en marte es dura”*). Es decir, muchas peticiones y pocos recursos:

- 80 peticiones
- 10 unidades de recursos
- 60 bases

Los rovers, aunque tienen una cantidad de combustible limitada, suelen tener suficiente para recorrer toda la red de bases al menos una vez. Debido a esto, es de esperar que al aumentar la cantidad de estos, aumente el tiempo de ejecución del programa, ya que este tendrá que hacer muchos más cálculos para encontrar una solución óptima.

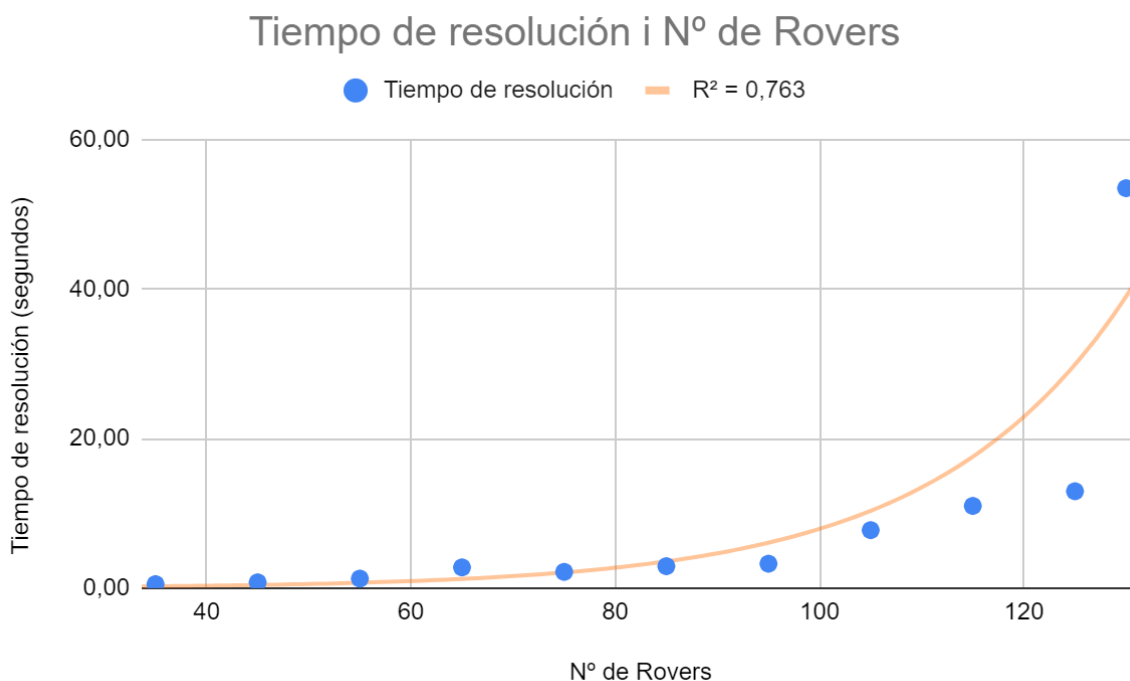


Figura 1: Gráfico del tiempo de resolución en función del número de rovers

En esta gráfica, podemos observar como aumenta el tiempo de ejecución al aumentar la cantidad de rovers disponibles, lo que concuerda con la hipótesis inicial. La línea de tendencia apunta claramente a una relación exponencial. Esto se debe a que todos los operadores involucran a los rovers, lo que causa que al aumentar la cantidad de estos, se aumente de forma exponencial la medida del árbol de decisiones.

## Prueba 2

En este caso, queremos estudiar la variación en el tiempo de ejecución del programa en función del número de peticiones. Hemos definido los siguientes valores basándonos en el mismo criterio que la prueba 1:

- 10 unidades de recursos
- 13 rovers
- 60 bases

La peculiaridad que tienen las peticiones, a diferencia de otros elementos del problema, es que la única implicación que tienen en los operadores es para terminar la rama de decisiones de un recurso. Esto causa que la cantidad de decisiones extra que se deben hacer al añadir una petición es mucho menor que al añadir, por ejemplo, un rover.

Sin embargo, las peticiones solo tienen 3 niveles de prioridad, por lo que es muy probable que al añadir varias peticiones, algún recurso tenga una petición del mismo nivel de prioridad o superior a una distancia menor (entendiendo por distancia el coste).

Debido a esto, suponemos que el tiempo de ejecución será menor al aumentar el número de peticiones.

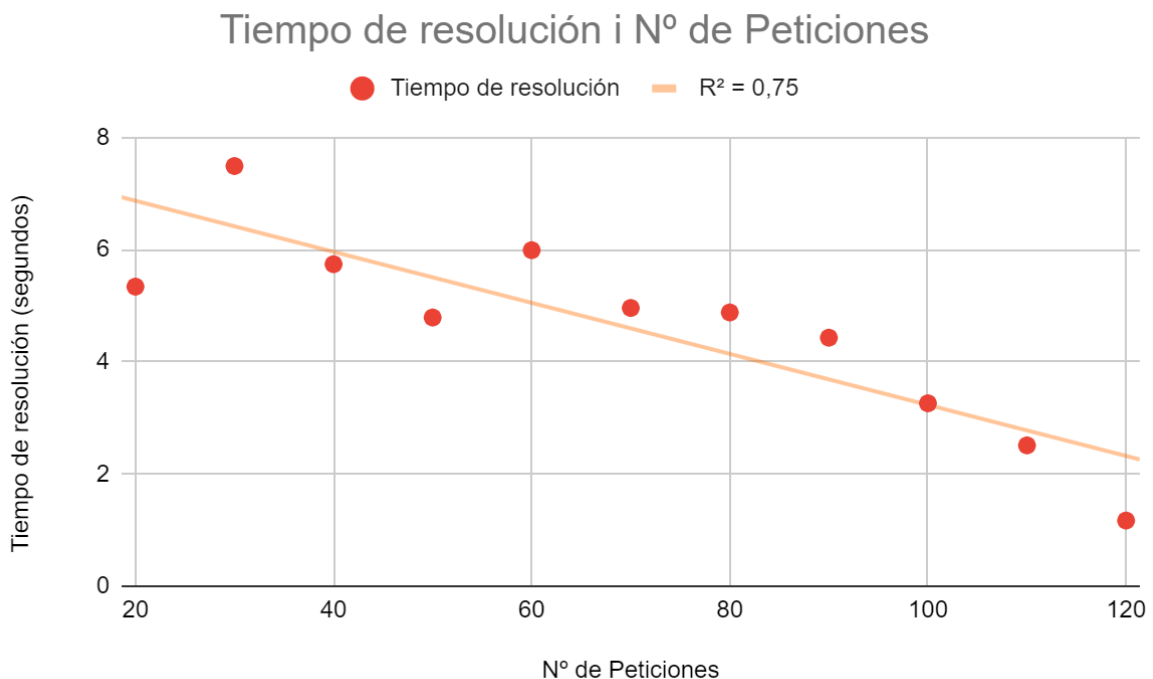


Figura 2: Gráfico del tiempo de resolución en función del número de peticiones

Al hacer el experimento, podemos observar que el tiempo de resolución decrece linealmente en función del número de peticiones, que concuerda con la hipótesis.

## Prueba 3

En este caso, queremos estudiar la variación en el tiempo de ejecución en función del número de recursos disponibles. Siguiendo el mismo criterio que en la prueba 1, hemos supuesto los siguientes datos:

- 30 peticiones
- 13 rovers
- 60 bases

En este caso, suponemos que al aumentar la cantidad de recursos disponibles, el coste en el tiempo de ejecución se verá incrementado, ya que aumentará la cantidad de datos que deberá procesar el programa.

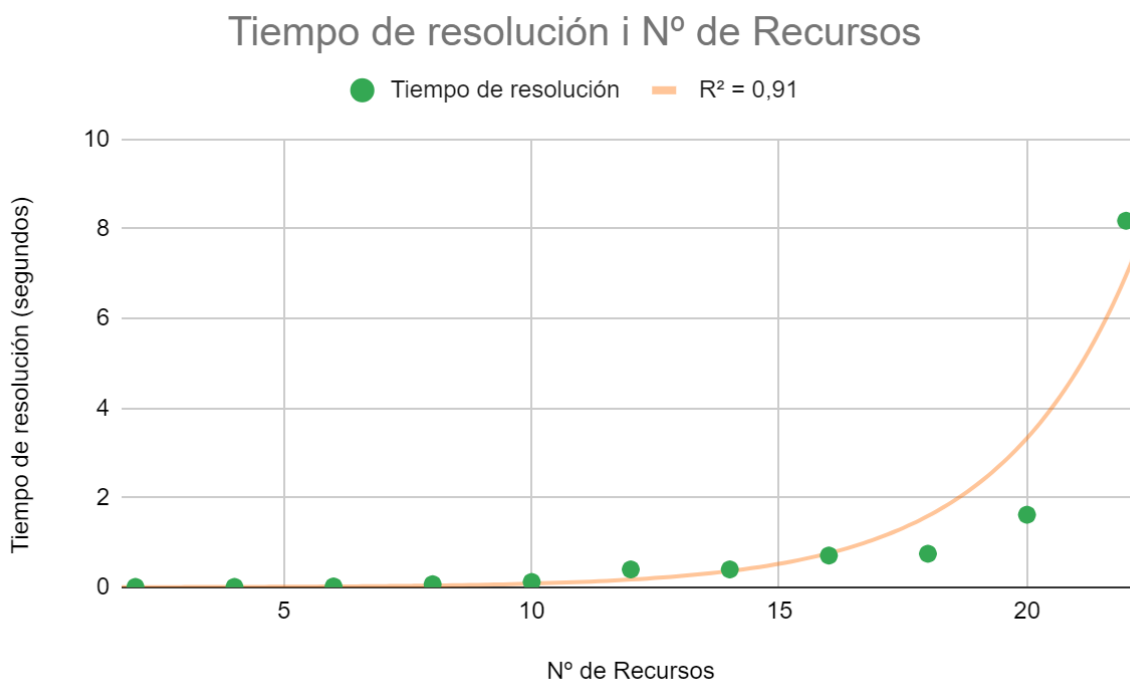


Figura 3: Gráfico del tiempo de resolución en función del número de recursos

Podemos ver que al aumentar la cantidad de recursos disponibles aumenta exponencialmente la cantidad de tiempo que requiere el programa para encontrar una solución. Esto se debe a que todas las acciones del programa envuelven de una forma u otra a los recursos. Esto provoca que al aumentar la cantidad de estos, aumente exponencialmente las posibles combinaciones de operadores que se pueden aplicar.



# Completado de los niveles

## Nivel Básico

En el nivel básico se nos pedía implementar un programa que creara un plan, teniendo en cuenta que habrá igual o más peticiones que recursos hay disponibles.

Para ello hemos creado las siguientes variables:

```
(:types
  recurso - object
  persona - recurso
  suministro - recurso
  base - object
  asentamiento - base
  almacen - base
  rover - object
)
```

Las cuales son las mismas para todas las versiones, y se han explicado anteriormente.

También hemos creado los siguientes predicados:

```
(:predicates
  (peticion ?rec - recurso ?baseDestino - base)
  (posicionRecurso ?rec - recurso ?b - base)
  (posicionRover ?rover - rover ?b - base)
  (basesAdyacentes ?b1 - base ?b2 - base)
  (transporta ?rover - rover ?rec - recurso)
  (entregado ?rec - recurso)
)
```

Los predicados *peticion*, *posicionRecurso*, *posicionRover* y *basesAdyacentes* se han explicado anteriormente. Por otro lado, el predicado *transporta* nos indica qué recurso está transportando el rover.

Finalmente, en esta versión tenemos dos acciones:

```
(:action mover
  :parameters (?rover - rover ?b1 - base ?b2 - base)
  :precondition (and
    (basesAdyacentes ?b1 ?b2)
    (posicionRover ?rover ?b1))
  :effect (and
    (posicionRover ?rover ?b2)
    ;;DEJAR RECURSOS
```

```

        (forall (?rec - recurso)
          (when (and
            (transporta ?rover ?rec)
            (peticion ?rec ?b2))
            (and
              (not (transporta ?rover ?rec))
              (entregado ?rec)
              (not (peticion ?rec ?b2))))))
    )

```

```

(:action coger
  :parameters (?rover - rover ?base - base ?rec - recurso)
  :precondition (and
    (posicionRecurso ?rec ?base)
    (posicionRover ?rover ?base)
    (not (entregado ?rec)))
  :effect (and
    (not (posicionRecurso ?rec ?base))
    (transporta ?rover ?rec))
)

```

La primera acción sirve para mover el rover y dejar todo lo pertinente en la base de destino. Recibe como parámetros un rover y dos bases, y su precondition es que el rover esté en la primera base, las dos bases sean adyacentes. Su efecto es que mueve el rover a la segunda base, y deja en esta todos los recursos que lleva actualmente, tal que exista una petición del recurso a dicha base.

La segunda acción sirve para coger recursos. Recibe como parámetro un rover, una base y un recurso. Su precondition es que el rover y el recurso estén en la base, y aún no se haya entregado el recurso. Su efecto es que el recurso deja de estar en la base y lo pasa a transportar el rover.

No se podría hacer una única acción que cogiera todos los recursos en algún momento, ya que de otra manera el programa no tendría la capacidad de repartir el trabajo entre los rovers.

## Extensión 1

Para esta versión se nos pedía añadir restricciones en la carga de los rovers. Ahora pueden llevar o bien un suministro, o bien un máximo de dos personas, no pudiendo mezclar tipos.

Hemos añadido la siguiente función:

```
(:functions
  (numPersonas ?rover - rover)
)
```

La cual se ha explicado anteriormente.

Hemos añadido también los siguientes predicados:

```
(transportaSuministro ?rover - rover ?sum - suministro)
(transportaPersona ?rover - rover ?per - persona)
(llevaSuministro ?rover - rover)
```

Estos tres predicados se han explicado anteriormente, ya que han llegado a la versión final.

Finalmente, hemos partido las acciones que teníamos en el nivel básico en las siguientes:

```
(:action mover
  :parameters (?rover - rover ?b1 - base ?b2 - base)
  :precondition (and
    (basesAdyacentes ?b1 ?b2)
    (posicionRover ?rover ?b1))
  :effect (and
    (posicionRover ?rover ?b2)
    (not (posicionRover ?rover ?b1)))
)
```

```
(:action cogerSuministro
  :parameters (?rover - rover ?pos - base ?sum - suministro)
  :precondition (and
    (posicionRover ?rover ?pos)
    (posicionRecurso ?sum ?pos)
    (not (entregado ?sum))
    (not (llevaSuministro ?rover))
    (= (numPersonas ?rover) 0))
  :effect (and
    (llevaSuministro ?rover)
    (transportaSuministro ?rover ?sum)
    (not (posicionRecurso ?sum ?pos)))
)
```

```
(:action cogerPersona
  :parameters (?rover - rover ?pos - base ?per - persona)
  :precondition (and
    (posicionRover ?rover ?pos)
```

```

        (posicionRecurso ?per ?pos)
        (not (entregado ?per))
        (not (llevaSuministro ?rover))
        (<= (numPersonas ?rover) 1))
:effect (and
        (increase (numPersonas ?rover) 1)
        (transportaPersona ?rover ?per)
        (not (posicionRecurso ?per ?pos)))
)

```

```

(:action dejarSuministro
  :parameters (?rover - rover ?pos - base ?sum - suministro)
  :precondition (and
    (posicionRover ?rover ?pos)
    (peticion ?sum ?pos)
    (transportaSuministro ?rover ?sum))
  :effect (and
    (not (transportaSuministro ?rover ?sum))
    (not (llevaSuministro ?rover))
    (not (peticion ?sum ?pos))
    (posicionRecurso ?sum ?pos)
    (entregado ?sum)))
)

```

```

(:action dejarPersona
  :parameters (?rover - rover ?pos - base ?per - persona)
  :precondition (and
    (posicionRover ?rover ?pos)
    (peticion ?per ?pos)
    (transportaPersona ?rover ?per)
    (> (numPersonas ?rover) 0))
  :effect (and
    (not (transportaPersona ?rover ?per))
    (decrease (numPersonas ?rover) 1)
    (not (peticion ?per ?pos))
    (posicionRecurso ?per ?pos)
    (entregado ?per)))
)

```

Todas estas acciones han llegado a la versión final, y, por tanto se han explicado anteriormente.

## Extensión 2

En esta versión se nos pedía añadir una limitación de combustible a los rovers. Cada movimiento entre dos bases adyacentes consume una unidad de combustible, y el rover requiere combustible para poder moverse. Se nos pedía, además, proporcionar otra subversión de esta extensión que permitiera minimizar el combustible gastado (equivalente a maximizar el restante).

Para poder hacer esto, hemos añadido las siguientes funciones:

```
(combustible ?rover - rover)
(sumaCombustible)
```

Las dos han sido explicadas anteriormente.

Además, se han modificado el operador *mover*:

```
(:action mover
  :parameters (?rover - rover ?b1 - base ?b2 - base)
  :precondition (and
    (basesAdyacentes ?b1 ?b2)
    (posicionRover ?rover ?b1)
    (> (combustible ?rover) 0))
  :effect (and
    (posicionRover ?rover ?b2)
    (not (posicionRover ?rover ?b1))
    (decrease (combustible ?rover) 1))
    (decrease (sumacombustible) 1)))
)
```

También está explicado anteriormente, el único cambio respecto a la extensión 1 es la comprobación del combustible en la precondition y la reducción de este en el efecto.

## Extensión 3

En esta última versión se nos pedía añadir al problema el concepto de prioridades. Una prioridad es un valor entre uno y tres asignado a una petición. El objetivo de las prioridades es maximizar la prioridad de las peticiones servidas.

Con tal de hacer esto, hemos añadido las siguientes funciones:

```
(prioridad ?rec - recurso ?baseDestino - base)
(sumaPrioridad)
```

Ambas explicadas ya anteriormente.

Además, se han modificado las funciones siguientes:

```
(:action dejarSuministro
  :parameters (?rover - rover ?pos - base ?sum - suministro)
  :precondition (and
    (posicionRover ?rover ?pos)
    (peticion ?sum ?pos)
    (transportaSuministro ?rover ?sum))
  :effect (and
    (not (transportaSuministro ?rover ?sum))
    (not (llevaSuministro ?rover))
    (not (peticion ?sum ?pos))
    (posicionRecurso ?sum ?pos)
    (entregado ?sum)
    (increase (sumaPrioridad) (prioridad ?sum ?pos)))
)
```

```
(:action dejarPersona
  :parameters (?rover - rover ?pos - base ?per - persona)
  :precondition (and
    (posicionRover ?rover ?pos)
    (peticion ?per ?pos)
    (transportaPersona ?rover ?per)
    (> (numPersonas ?rover) 0))
  :effect (and
    (not (transportaPersona ?rover ?per))
    (decrease (numPersonas ?rover) 1)
    (not (peticion ?per ?pos))
    (posicionRecurso ?per ?pos)
    (entregado ?per)
    (increase (sumaPrioridad) (prioridad ?per ?pos)))
)
```

Ambas funciones están explicadas anteriormente, los únicos cambios respecto a la extensión 2 son que, una vez dejamos un recurso, sumamos a *sumaPrioridad* la prioridad asociada a esa petición, para así tener un valor que maximizar.

# Conclusión

En esta práctica, hemos analizado un problema de planificar los movimientos de rovers para poder transportar distintos recursos en Marte, y hemos modelado los elementos a partir de objetos, predicados, funciones y operadores. Hemos ido desarrollando nuevas versiones por cada extensión que se indicaba en la práctica, con un diseño incremental.

Durante el desarrollo de esta práctica, hemos notado muchas similitudes con la práctica anterior, de sistemas basados en el conocimiento. Al crear el sistema, se sigue una metodología parecida, con un prototipado rápido y añadiendo extensiones incrementalmente. Además, en ambos casos llegamos a una solución partiendo de un estado inicial, y aplicando operadores con restricciones y efectos. Aun así existen ciertas diferencias, como por ejemplo que en la práctica anterior el criterio para escoger qué operador aplicar era distinto, y en esta se guarda el camino entero para llegar a una solución.

En el análisis de tiempo de ejecución que hemos realizado hemos encontrado la debilidad de los sistemas basados en PDDL: la mayoría de curvas del tiempo con relación a la cantidad de objetos son exponenciales. Esto nos dificulta usar esta IA para proyectos muy extensos, pero pensamos que sigue siendo extremadamente útil para proyectos de planificación sencillos sobre todo por la facilidad y rapidez al escribir el código.