

TRABAJO TEMA 6

LUIS ALBERTO ÁLVAREZ ZAVALETA

Definición del problema

Para poder definir este problema se ha utilizado la plantilla de DomainDriverLogic como base, se han hecho algunos cambios a su estructura, en primer lugar a los tipos.

```
Show hierarchy
(:types
  city - object
  house station airport - location
  locatable transport - movables
  driver parcel - locatable
  truck plane train - transport
)
```

Hemos creado una estructura que nos permite separar paquetes y conductores, en un grupo los cuales son elementos que podrán luego estar dentro de los elementos de transport.

Pero dado que ambos elementos pueden estar en algún lugar, me he decidido por esta estructura. Por último será necesario saber en que lugar esta ubicada la ciudad, esto servirá para poder implementar la restricción que impide a los camiones viajar a otras ciudades.

```
(:predicates
  (located ?loc - location ?city - city) 4
  (at ?obj - movables ?loc - location)
  (path ?x ?y - location) 4
  (loaded ?plc - parcel ?trs - transport) 3 3 3
  (driving ?d - driver ?v - transport) 2 1 1
)
```

Se han implementado la misma estructura que en la plantilla original a la hora de mover paquetes, el poder cargar y descargar un paquete, así como ser posible para un conductor ser posible subirse o bajarse del camión.

Por último, en la definición del problema se deberán definir los caminos. Esto es las conexiones entre cada localización entre las ciudades, así como unir todas las estaciones de tren y todos los aeropuertos que estén conectados.

Para poder comprobar el tiempo que cada uno de los transportes se tardan en llegar a un lugar tenemos las 4 primeras funciones. Además, dado que cada camión tiene una capacidad y el paquete un peso, se deberán definir en el problema.

```
(:functions
  (time-to-walk ?loc ?loc1 - location) 1
  (time-to-drive ?loc ?loc1 - location) 1
  (time-to-fly ?loc ?loc1 - airport) 1
  (time-to-train ?loc ?loc1 - station) 1
  (capacity-weight ?truck - truck) 1 1 1
  (parcel-weight ?pkg - parcel) 3
)
```

Entre algunos métodos interesantes a analizar se encuentra por ejemplo el método de cargar al camión. En este caso podemos observar que Entre las condiciones iniciales se encuentra que el camión se encuentra en la localización durante la carga y que el paquete tambien se encuentra en ella, además de comprobar que la carga del paquete cabe en el camión. En caso de que se cumpla el camión pasará a estar cargado en el paquete.

```
(:durative-action LOAD-TRUCK
  :parameters (?pkg - parcel ?truck - truck ?loc - location)
  :duration (= ?duration 1)
  :condition (and
    (over all (at ?truck ?loc))
    (at start (at ?pkg ?loc))
    (at start (>= (capacity-weight ?truck) (parcel-weight ?pkg)))
  )
  :effect (and
    (at start (not (at ?pkg ?loc)))
    (at end (loaded ?pkg ?truck))
    (at end (decrease (capacity-weight ?truck) (parcel-weight ?pkg)))
  )
)
```

Otro método que es interesante de analizar es el método de caminar o conducir un camión, en este caso se necesitará asegurarnos que tanto el camión, la localización de destino y la de inicio se encuentran en la misma ciudad. Además, un conductor deberá conducir el camión durante todo el trayecto, así como la condición de que el camino debe existir durante la duración del viaje. Con esto el camión se desplazará de un lugar a otro.

```
(:durative-action DRIVE-TRUCK
  :parameters (?truck - truck ?loc-from - location ?loc-to - location ?driver - driver ?city - city)
  :duration (= ?duration (time-to-drive ?loc-from ?loc-to))
  :condition (and
    (at start (at ?truck ?loc-from))
    (at start (located ?loc-from ?city))
    (at start (located ?loc-to ?city))
    (over all (driving ?driver ?truck))
    [(over all) (path ?loc-from ?loc-to)]
  )
  :effect (and (at start (not (at ?truck ?loc-from)))
    (at end (at ?truck ?loc-to)))
  )
)
```

Por último, podemos observar como es el método que permite al conductor subir al camión, esto será necesario para que se pueda utilizar el método anterior.

```
(:durative-action BOARD-TRUCK
  :parameters (?driver - driver ?truck - truck ?loc - location)
  :duration (= ?duration 1)
  :condition (and
    (over all (at ?truck ?loc))
    (at start (at ?driver ?loc))
  )
  :effect (and
    (at start (not (at ?driver ?loc)))
    (at end (driving ?driver ?truck))
  )
)
```

A diferencia del método de mover el camión, los aviones y las estaciones de tren pueden moverse a cualquiera de ellas, siempre que haya un path definido en el problema que los conecte, mediante el método siguiente.

```
(:durative-action MOVE-TRAIN
  :parameters (
    | ?train - train ?loc-from - station ?loc-to - station
  )
  :duration (= ?duration (time-to-train ?loc-from ?loc-to))
  :condition (and
    (at start (at ?train ?loc-from))
    (over all (path ?loc-from ?loc-to))
  )
  :effect (and
    (at start (not (at ?train ?loc-from)))
    (at end (at ?train ?loc-to))
  )
)
```

De la misma forma que en el caso del camión es posible cargar y descargar el paquete, en el caso de los trenes y aviones no se aplican restricciones de peso, aunque sería posible añadirlos

```
(:durative-action LOAD-TRAIN
  :parameters (?pkg - parcel ?train - train ?loc - station)
  :duration (= ?duration 1)
  :condition (and
    (over all (at ?train ?loc))
    (at start (at ?pkg ?loc))
  )
  :effect (and
    (at start (not (at ?pkg ?loc)))
    (at end (loaded ?pkg ?train))
  )
)
```

Evaluación del problema

En cuanto a especificaciones del problema. Son las que se han planteado

En primer lugar se deben definir los vehículos y pesos de cada uno de los camiones , así como los paquetes, en este caso he decidido que cada camión pueda llevar un peso arbitrario definido por mi pero se puede cambiar en la definición del problema.

```
;;Definir la localización de los paquetes y su peso
(at parcel1 home2)
(at parcel2 home1)
(= (parcel-weight parcel1) 15)
(= (parcel-weight parcel2) 20)

;Definir de los vehículos y sus pesos
(at truck1 station1)
(at truck2 airport1)
(at truck3 airport2)
(at truck4 home3)

(= (capacity-weight truck1) 30)
(= (capacity-weight truck2) 40)
(= (capacity-weight truck3) 20)
(= (capacity-weight truck4) 40)
```

Además se necesitan crear los caminos que pueden seguir los camiones .Además de asignar el peso que debe realizar el camión . Para simplificar el problema todas las acciones originales tendrán un peso de 1 unidad de tiempo.

```
;City1
(path home1 station1)
(path home1 station2)
(path home1 airport1)
(= (time-to-walk home1 station1) 1)
(= (time-to-walk home1 station2) 1)
(= (time-to-walk home1 airport1) 1)
(= (time-to-drive home1 station1) 1)
(= (time-to-drive home1 station2) 1)
(= (time-to-drive home1 airport1) 1)
```

Por ultimo es necesario definir la meta y en este caso la métrica que se quiere utilizar , en nuestro caso es minimizar el tiempo total que tardará en realizarse el plan.

```
(:goal (and
  (at parcel1 home4)
  (at parcel2 home4)

))

(:metric minimize (total-time))
```

Usando -n 1, podemos apreciar que el camión nos da una solución de 16 unidades de tiempo. He observado que el programa lpg-td tiene.

```
Plan computed:
  Time: (ACTION) [action Duration; action Cost]
0.0000: (BOARD-TRUCK DRIVER3 TRUCK4 HOME3) [D:1.0000; C:0.1000]
0.0000: (MOVE-TRAIN TRAIN2 STATION4 STATION1) [D:1.0000; C:0.1000]
0.0000: (WALK DRIVER1 STATION2 AIRPORT1 CITY1) [D:1.0000; C:0.1000]
0.0000: (MOVE-TRAIN TRAIN1 STATION2 STATION4) [D:1.0000; C:0.1000]
0.0000: (WALK DRIVER2 HOME2 AIRPORT2 CITY2) [D:1.0000; C:0.1000]
1.0000: (DRIVE-TRUCK TRUCK4 HOME3 STATION4 DRIVER3 CITY3) [D:1.0000; C:0.1000]
1.0000: (BOARD-TRUCK DRIVER1 TRUCK2 AIRPORT1) [D:1.0000; C:0.1000]
1.0000: (MOVE-TRAIN TRAIN1 STATION4 STATION3) [D:1.0000; C:0.1000]
1.0000: (BOARD-TRUCK DRIVER2 TRUCK3 AIRPORT2) [D:1.0000; C:0.1000]
2.0000: (DRIVE-TRUCK TRUCK2 AIRPORT1 STATION2 DRIVER1 CITY1) [D:1.0000; C:0.1000]
2.0000: (DRIVE-TRUCK TRUCK3 AIRPORT2 HOME2 DRIVER2 CITY2) [D:1.0000; C:0.1000]
3.0000: (DRIVE-TRUCK TRUCK2 STATION2 HOME1 DRIVER1 CITY1) [D:1.0000; C:0.1000]
3.0000: (LOAD-TRUCK PARCEL1 TRUCK3 HOME2) [D:1.0000; C:0.1000]
4.0000: (LOAD-TRUCK PARCEL2 TRUCK2 HOME1) [D:1.0000; C:0.1000]
4.0000: (DRIVE-TRUCK TRUCK3 HOME2 STATION3 DRIVER2 CITY2) [D:1.0000; C:0.1000]
5.0000: (DRIVE-TRUCK TRUCK2 HOME1 STATION1 DRIVER1 CITY1) [D:1.0000; C:0.1000]
5.0000: (UNLOAD-TRUCK PARCEL1 TRUCK3 STATION3) [D:1.0000; C:0.1000]
6.0000: (UNLOAD-TRUCK PARCEL2 TRUCK2 STATION1) [D:1.0000; C:0.1000]
6.0000: (LOAD-TRAIN PARCEL1 TRAIN1 STATION3) [D:1.0000; C:0.1000]
7.0000: (LOAD-TRUCK PARCEL2 TRUCK1 STATION1) [D:1.0000; C:0.1000]
7.0000: (MOVE-TRAIN TRAIN1 STATION3 STATION4) [D:1.0000; C:0.1000]
8.0000: (UNLOAD-TRUCK PARCEL2 TRUCK1 STATION1) [D:1.0000; C:0.1000]
8.0000: (UNLOAD-TRAIN PARCEL1 TRAIN1 STATION4) [D:1.0000; C:0.1000]
9.0000: (LOAD-TRAIN PARCEL2 TRAIN2 STATION1) [D:1.0000; C:0.1000]
10.0000: (MOVE-TRAIN TRAIN2 STATION1 STATION4) [D:1.0000; C:0.1000]
11.0000: (UNLOAD-TRAIN PARCEL2 TRAIN2 STATION4) [D:1.0000; C:0.1000]
11.0000: (LOAD-TRAIN PARCEL1 TRAIN2 STATION4) [D:1.0000; C:0.1000]
12.0000: (LOAD-TRUCK PARCEL2 TRUCK4 STATION4) [D:1.0000; C:0.1000]
12.0000: (UNLOAD-TRAIN PARCEL1 TRAIN2 STATION4) [D:1.0000; C:0.1000]
13.0000: (LOAD-TRUCK PARCEL1 TRUCK4 STATION4) [D:1.0000; C:0.1000]
14.0000: (DRIVE-TRUCK TRUCK4 STATION4 HOME4 DRIVER3 CITY3) [D:1.0000; C:0.1000]
15.0000: (UNLOAD-TRUCK PARCEL2 TRUCK4 HOME4) [D:1.0000; C:0.1000]
15.0000: (UNLOAD-TRUCK PARCEL1 TRUCK4 HOME4) [D:1.0000; C:0.1000]
```

En cuanto a tiempos en el problema base usando – quality obtenemos una duración de 13 unidades de tiempo pero por alguna razón no lpg-td no devuelve el plan, solo su duración:

```
Plan computed:
  Time: (ACTION) [action Duration; action Cost]

Solution found:
Total time:      0.02
Search time:     0.00
Actions:         26
Execution cost:  2.60
Duration:        13.000
Plan quality:    13.000
  Plan file:     plan_problemaBase.SOL

Do not use option -quality for better solutions.

C:\Users\alber\OneDrive\Documentos\UPV\2MU\LSE\Domain>
```

he probado usando el archivo mips-xxl en Linux y en este caso el problema devuelve correctamente la solución de calidad de tiempo 13 unidades.

```
; ParsingTime
; NrActions 27
; MakeSpan 13.05
; MetricValue 0.00
; PlaningTechnique best-first
0.00: (WALK DRIVER2 HOME2 AIRPORT2 CITY2 ) [1.00]
0.00: (WALK DRIVER1 STATION2 AIRPORT1 CITY1 ) [1.00]
0.00: (BOARD-TRUCK DRIVER3 TRUCK4 HOME3 ) [1.00]
0.00: (MOVE-TRAIN TRAIN2 STATION4 STATION3 ) [1.00]
1.00: (DRIVE-TRUCK TRUCK4 HOME3 STATION4 DRIVER3 CITY3 ) [1.00]
1.01: (BOARD-TRUCK DRIVER2 TRUCK3 AIRPORT2 ) [1.00]
1.01: (BOARD-TRUCK DRIVER1 TRUCK2 AIRPORT1 ) [1.00]
2.01: (DRIVE-TRUCK TRUCK2 AIRPORT1 HOME1 DRIVER1 CITY1 ) [1.00]
2.01: (DRIVE-TRUCK TRUCK3 AIRPORT2 HOME2 DRIVER2 CITY2 ) [1.00]
3.01: (LOAD-TRUCK PARCEL2 TRUCK2 HOME1 ) [1.00]
3.01: (LOAD-TRUCK PARCEL1 TRUCK3 HOME2 ) [1.00]
4.01: (DRIVE-TRUCK TRUCK2 HOME1 STATION1 DRIVER1 CITY1 ) [1.00]
4.01: (DRIVE-TRUCK TRUCK3 HOME2 STATION3 DRIVER2 CITY2 ) [1.00]
5.01: (DRIVE-TRUCK TRUCK2 STATION1 STATION2 DRIVER1 CITY1 ) [1.00]
5.01: (UNLOAD-TRUCK PARCEL1 TRUCK3 STATION3 ) [1.00]
6.01: (UNLOAD-TRUCK PARCEL2 TRUCK2 STATION2 ) [1.00]
6.02: (LOAD-TRAIN PARCEL1 TRAIN2 STATION3 ) [1.00]
7.02: (LOAD-TRAIN PARCEL2 TRAIN1 STATION2 ) [1.00]
7.02: (MOVE-TRAIN TRAIN2 STATION3 STATION4 ) [1.00]
8.02: (MOVE-TRAIN TRAIN1 STATION2 STATION4 ) [1.00]
8.02: (UNLOAD-TRAIN PARCEL1 TRAIN2 STATION4 ) [1.00]
9.02: (UNLOAD-TRAIN PARCEL2 TRAIN1 STATION4 ) [1.00]
9.03: (LOAD-TRUCK PARCEL1 TRUCK4 STATION4 ) [1.00]
10.04: (LOAD-TRUCK PARCEL2 TRUCK4 STATION4 ) [1.00]
11.04: (DRIVE-TRUCK TRUCK4 STATION4 HOME4 DRIVER3 CITY3 ) [1.00]
12.04: (UNLOAD-TRUCK PARCEL1 TRUCK4 HOME4 ) [1.00]
12.05: (UNLOAD-TRUCK PARCEL2 TRUCK4 HOME4 ) [1.00]
```

Modelado de problema real

Para modelar el problema he usado GraphHopper y creado el fichero problemaDatosReales , en el he cambiado las ubicaciones genéricas por localizaciones en España , En concreto las ciudades y las localizaciones quedan de la siguiente manera. Luego he agregado manualmente los tiempos aproximados que daba la API para los viajes entre lugares.

```
Valencia - city
CalleArzobispoOlaechea - house
EstacionJoaquinSorolla - station
EstacionDelNorte - station
AeropuertoValencia - airport

Malaga - city
CalleCondeBelalcazar - house
AeropuertoCostaSol - airport
EstacionMariaZambrano - station

Madrid - city
CalleEdgarNeville - house
CalleDivinoValles - house
EstacionPuertaAtocha - station
```

Este es un ejemplo para los viajes desde la Casa1 en Valencia hasta las distintas estaciones y el aeropuerto., el tiempo esta expresado en minutos de duración.

```
;Valencia
(path CalleArzobispoOlaechea EstacionJoaquinSorolla)
(path CalleArzobispoOlaechea EstacionDelNorte)
(path CalleArzobispoOlaechea AeropuertoValencia)
(= (time-to-walk CalleArzobispoOlaechea EstacionJoaquinSorolla) 33)
(= (time-to-walk CalleArzobispoOlaechea EstacionDelNorte) 39)
(= (time-to-walk CalleArzobispoOlaechea AeropuertoValencia) 133)

(= (time-to-drive CalleArzobispoOlaechea EstacionJoaquinSorolla) 6)
(= (time-to-drive CalleArzobispoOlaechea EstacionDelNorte) 9)
(= (time-to-drive CalleArzobispoOlaechea AeropuertoValencia) 16)
```


Para este problema he usado directamente mips-xxl, podemos observar que la solución que nos devuelve es de 313 minutos o unas 5 horas y 13 minutos, junto al plan para llevar los 2 paquetes a casa4 el cuales es Calle Vino Vidales.

```
; ParsingTime
; NrActions 29
; MakeSpan 313.03
; MetricValue 0.00
; PlaningTechnique best-first
0.00: (WALK DRIVER2 CALLECONDEBELALCAZAR AEROPUERTOCOSTASOL MALAGA ) [125.00]
0.00: (WALK DRIVER1 ESTACIONDELNORTE AEROPUERTOVALENCIA VALENCIA ) [125.00]
0.00: (BOARD-TRUCK DRIVER3 TRUCK4 CALLEEDGARNEVILLE ) [1.00]
0.00: (MOVE-TRAIN TRAIN2 ESTACIONPUERTAATOCHA ESTACIONMARIAZAMBRANO ) [135.00]
1.00: (DRIVE-TRUCK TRUCK4 CALLEEDGARNEVILLE ESTACIONPUERTAATOCHA DRIVER3 MADRID ) [13.00]
125.01: (BOARD-TRUCK DRIVER2 TRUCK3 AEROPUERTOCOSTASOL ) [1.00]
125.01: (BOARD-TRUCK DRIVER1 TRUCK2 AEROPUERTOVALENCIA ) [1.00]
126.01: (DRIVE-TRUCK TRUCK2 AEROPUERTOVALENCIA CALLEARZOBISPOOLAECHEA DRIVER1 VALENCIA ) [16.00]
126.01: (DRIVE-TRUCK TRUCK3 AEROPUERTOCOSTASOL CALLECONDEBELALCAZAR DRIVER2 MALAGA ) [17.00]
142.01: (LOAD-TRUCK PARCEL2 TRUCK2 CALLEARZOBISPOOLAECHEA ) [1.00]
143.01: (DRIVE-TRUCK TRUCK2 CALLEARZOBISPOOLAECHEA ESTACIONJOAQUINSOROLLA DRIVER1 VALENCIA ) [6.00]
143.01: (LOAD-TRUCK PARCEL1 TRUCK3 CALLECONDEBELALCAZAR ) [1.00]
144.01: (DRIVE-TRUCK TRUCK3 CALLECONDEBELALCAZAR ESTACIONMARIAZAMBRANO DRIVER2 MALAGA ) [12.00]
149.01: (DRIVE-TRUCK TRUCK2 ESTACIONJOAQUINSOROLLA ESTACIONDELNORTE DRIVER1 VALENCIA ) [4.00]
153.01: (UNLOAD-TRUCK PARCEL2 TRUCK2 ESTACIONDELNORTE ) [1.00]
154.02: (LOAD-TRAIN PARCEL2 TRAIN1 ESTACIONDELNORTE ) [1.00]
155.02: (MOVE-TRAIN TRAIN1 ESTACIONDELNORTE ESTACIONPUERTAATOCHA ) [133.00]
156.01: (UNLOAD-TRUCK PARCEL1 TRUCK3 ESTACIONMARIAZAMBRANO ) [1.00]
157.02: (LOAD-TRAIN PARCEL1 TRAIN2 ESTACIONMARIAZAMBRANO ) [1.00]
158.02: (MOVE-TRAIN TRAIN2 ESTACIONMARIAZAMBRANO ESTACIONPUERTAATOCHA ) [135.00]
288.02: (UNLOAD-TRAIN PARCEL2 TRAIN1 ESTACIONPUERTAATOCHA ) [1.00]
293.02: (UNLOAD-TRAIN PARCEL1 TRAIN2 ESTACIONPUERTAATOCHA ) [1.00]
294.03: (LOAD-TRUCK PARCEL1 TRUCK4 ESTACIONPUERTAATOCHA ) [1.00]
295.03: (DRIVE-TRUCK TRUCK4 ESTACIONPUERTAATOCHA CALLEDIVINOVALLES DRIVER3 MADRID ) [5.00]
300.03: (UNLOAD-TRUCK PARCEL1 TRUCK4 CALLEDIVINOVALLES ) [1.00]
301.03: (DRIVE-TRUCK TRUCK4 CALLEDIVINOVALLES ESTACIONPUERTAATOCHA DRIVER3 MADRID ) [5.00]
306.03: (LOAD-TRUCK PARCEL2 TRUCK4 ESTACIONPUERTAATOCHA ) [1.00]
307.03: (DRIVE-TRUCK TRUCK4 ESTACIONPUERTAATOCHA CALLEDIVINOVALLES DRIVER3 MADRID ) [5.00]
312.03: (UNLOAD-TRUCK PARCEL2 TRUCK4 CALLEDIVINOVALLES ) [1.00]
```

Conclusiones

El problema ha sido interesante de resolver, modelar el problema con PDDL ha sido interesante para comprender el desarrollo técnico que se debe plantear para resolver un problema de logística de entrega de paquetes.

Entre algunas mejoras al modelo planteado que se me han ocurrido para ampliar el modelo que he planteado se encuentran las siguientes:

De la misma forma que se plantean capacidades y pesos en los paquetes para los camiones, plantear capacidades para trenes y aviones

Añadir nuevos vehículos, en concreto vehículos, que permitan a los conductores moverse más rápido hacia los camiones dado que en el problema real parte de ese tiempo se dedicaba a moverse a pie hacia los camiones porque no había otra opción.

Asignar precios a desplazamientos y tenerlo en cuenta como otra posible métrica a utilizar.

Añadir combustible a los camiones e implementar lugares donde puedan recargarlos , esto puede permitir añadir como metas el que los camiones deben tener cierto nivel de gasolina al final del problema.