**PRÀCTICA DE PLANIFICACIÓ**

*Jinheng Lin, Xin Lu i Hang Yao*

### Intel·ligència Artificial. Curs 2022-2023 Q1

**ÍNDEX**

1. **Descripció del problema**
2. **Nivell bàsic**
   1. Domini
   2. Problema
   3. Desenvolupament dels models
   4. Jocs de prova
3. **Extensió 1**
   1. Domini
   2. Problema
   3. Desenvolupament dels models
   4. Jocs de prova
4. **Extensió 2**
   1. Domini
   2. Problema
   3. Desenvolupament dels models
   4. Jocs de prova
5. **Extensió 3**
   1. Domini
   2. Problema
   3. Desenvolupament dels models
   4. Jocs de prova
6. **Conclusió**
7. **DESCRIPCIÓ DEL PROBLEMA**

En aquesta pràctica centrem en l’estudi de la planificació del problema, en què consisteix solucionar un problema concret a partir del disseny d’un curs d’acció que satisfà un cert objectiu. Dit d’una altra manera, és un conjunt d’accions definides per l’usuari que permet conduir un estat inicial a un estat final (objectiu que volem aconseguir). Doncs, es tracta de problemes de síntesi, en el qual s’ha de construir una solució basada en el coneixement del domini (l’estat de l’entorn, les acciones disponibles en cada moment i l’objectiu que es desitja a obtenir).

Aquest treball és un bon exemple per aplicar la resolució d’un problema simple d’assignació que es pot modelitzar com un problema de planificació per guiar adequadament la construcció de l’assignació. El problema consisteix en la distribució de P personals i S subministraments dels E emmagatzems als P peticions fets per A assentaments mitjançant els desplaçaments de R *rovers*, tals que només es poden desplaçar entre les bases que estan connexos (magatzems i assentaments són bases). Explicada d’una manera més clara seria planificar els desplaçaments dels *rovers*, que es troben aparcats a una base, fins a recollir els subministraments i personals necessaris per deixar-los on s’ha fet les peticions (cada petició és una unitat de suministre o una persona).

Un punt important del problema és que els *rovers* només poden moure entre les bases que estan connexos (definim un graf connex de bases a l’estat inicial). L’altre punt destacable és que sempre hi ha més peticions de subministraments i personals de les quals disposen als emmagatzematges, per tant, el nombre de peticions de cada cosa és igual o major del qual que està disponible.

En aquesta pràctica hem considerat la versió bàsica del problema i tres extensions proposades, de forma incremental. Es pot desenvolupar mitjançant un model expressat en PDDL per cada un dels problemes considerats, analitzant els elements necessaris i la diferència entre ells, i resolem utilitzant l’eina de planificació Fast Forward. No sempre ens ofereix una solució òptima, però si les aproximades, ja que fa servir una recerca amb BFS.

Els següents apartats, expliquem detalladament en què consisteix el problema de les diferents versions, els seus dominis i el desenvolupament del model. A més d’això, també proposem una possible solució d’un joc de prova adequat per cada problema, per tal de poder entendre millor la solució dels problemes en cada cas.

1. **NIVELL BÀSIC**

El problema d’aquest apartat es tracta de fer un planificador de realitzar el transport de tots els subministraments i personals dels magatzems per cobrir el màxim de les peticions amb els que hi disposen. Especifiquem que no hi ha cap restricció en la quantitat de càrrega de subministraments i personals pel rover i es pot transportar tots dos a la vegada.

* 1. **Domini**

A partir de l’enunciat de la pràctica podem definir els diferents elements del domini, que els instanciem al següent:

* Variables:
  + rover: tipus que indica que l’objecte és un rover.
  + peticion: tipus que indica que l’objecte és una petició.
  + base: tipus que indica que l’objecte és una base, pot ser un assentament o un magatzem.
  + asentamiento: tipus que indica que la base és un assentament.
  + almacen: tipus que indica que la base és un magatzem.
* Predicats:
  + (esta ?r - rover ?b - base): indica que el rover r està aparcat en la base b.
  + (conectado ?b1 - base ?b2 - base): indica que la base b1 està connectada amb la base b2.
  + (disponible ?b - base): indica que la base b disposa d’algún subministrament o personal.
  + (no-disponible ?b - base): indica que la base b no disposa de cap subministrament ni personal.
  + (destinoPeticion ?p - peticion ?a - asentamiento): indica que l’assentament a ha fet la petició p.
  + (vacio ?r - rover): indica que el rover r està buit, no transporta res.
  + (peticion-dePersonal ?p - peticion): indica que la petició p és una petició que demana un personal.
  + (peticion-deSuministro ?p - peticion): indica que la petició p és una petició que demana un subministrament.
  + (no-hecho ?p - peticion): indica que la petició p no s’ha entregat.
* Funcions:
  + (disponiblePersonal ?a - base): la quantitat de personals que té la base b.
  + (disponibleSuministro ?a - base): la quantitat de subministraments que té la base b.
  + (numPersonal ?r - rover): la quantitat de personals que transporta el rover r.
  + (numSuministro ?r - rover): la quantitat de subministraments que transporta el rover r.
* Accions:
  + mover(?r - rover ?b1 - base ?b2 - base): moure el rover r de la base b1 a la base b2.
    - Precondicions: el rover r està en la base b1 i hi ha connectivitat entre les bases b1 i b2.
  + cargarPersonal(?r - rover ?a - asentamiento): carrega tots els personals que disposa l’assentament a al rover r.
    - Precondicions: el rover r està en l’assentament a i aquest és disponible.
  + cargarSuministro(?r - rover ?a - almacen): carrega tots els subministraments que disposa el magatzem a al rover r.
    - Precondicions: el rover r està en l’assentament a i aquest és disponible.
  + descargarRover(?r - rover ?a - asentamiento ?p - peticion): descarrega del rover r el que demana la petició p a l’assentament a, on s’ha fet la petició.
    - Precondicions: no s’ha fet la petició p, el rover r està en l’assentament a, el rover r no està buit i la destinació de la petició p és l’assentament a.
  + revisaEstadoAlmacen(?a - almacen): actualitza la disponibilitat de subministraments del magatzem a.
  + revisaEstadoAsentamiento(?a - asentamiento): actualitza la disponibilitat de personals de l’assentament.
  + revisaEstadoRover(?r - rover): actualitza l’estat del rover, està buit o no.
  1. **Problema**

Tenint en compte el domini especificat anteriorment, plantegem un problema amb les següents característiques:

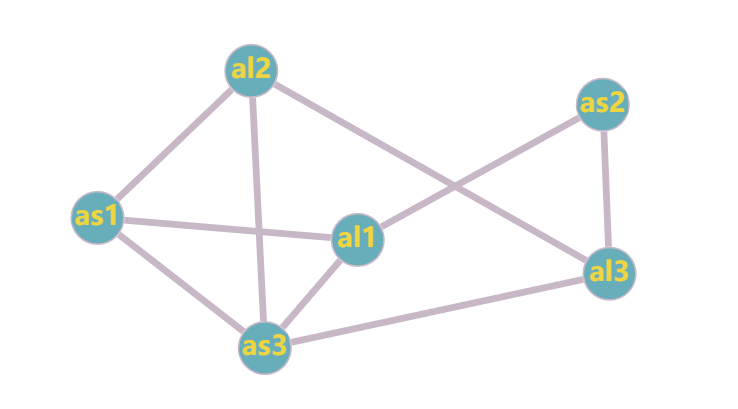
* Objectes: definim els objectes necessaris pel problema, que són les bases, les peticions i els rovers.
* Estat inicial: inicialitzem les connectivitats entre les diferents bases, on estan aparcats els rovers, la disponibilitat dels magatzems i les destinacions de les peticions i el seu tipus.
* Estat final: l’objectiu que volem arribar és buidar tot que hi disposen les bases i, també, els rovers per tal de cobrir el màxim de les peticions.
  1. **Desenvolupament dels models**

Pel desenvolupament d’aquest problema, hem optat per un disseny basat en un prototip seguint el nivell bàsic del problema. En aquest prototip, hem afegit els elements més principals del problema, mencionades als subapartats anteriors.

* 1. **Jocs de prova**
* Joc de prova 1:

Aquest joc de prova disposa d’un conjunt de 6 bases, 3 de tipus assentament i 3 de tipus magatzem, 2 rovers i 5 peticions.

Per ells, hem definit la connectivitat entre les bases de la següent manera:



on asX són els assentaments i alX són els magatzems.

Les disponibilitats dels magatzems són:

* al1: aquest magatzem disposa només d’1 unitat de subministrament.
* al2: aquest magatzem disposa només d’1 personal.
* al3: aquest magatzem disposa d’1 unitat de subministrament i d’1 personal.

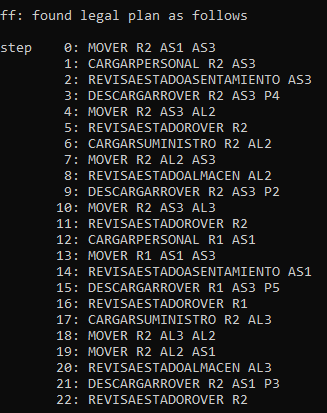
Per la part dels rovers, tots dos rovers r1 i r2 estan aparcats en l’assentament as1 i, òbviament, estan buits inicialment.

Per la part de les peticions, tenim que:

* p1: de tipus personal amb destinació d’as2.
* p2: de tipus subministrament amb destinació d’as3.
* p3: de tipus subministrament amb destinació d’as1.
* p4: de tipus personal amb destinació d’as3.
* p5: de tipus personal amb destinació d’as3.

Amb l’anterior inicialització volem trobar una planificació per tal de complir el nostre objectiu plantejat anteriorment. Esperem una solució vàlida, ja que el graf que connecta les bases resulta connex. Així que el rover pot arribar a qualsevol base des d’una base determinada.

I la solució generada és tal com havíem esperat, els rovers han pogut transportar tot el que disposa els magatzems i buidar també ells mateixos. En el següent mostrem la part del pla obtingut en l’output. El complet l’hem ajuntat en el fitxer [output1\_problema\_basic.txt] a part per poder veure amb més claredat.



* Joc de prova 2:

1. **EXTENSIÓ 1**

Com a extensió del nivell bàsic, a part que s’ha de fer el mateix que aquell, en aquesta secció s’afegeix la restricció que cada rover només pot tenir un màxim de dues persones o una càrrega de subministraments (no es poden barrejar persones i subministraments).

* 1. **Domini**

Per satisfer la nova restricció d’aquesta extensió, hem fet modificacions a partir de la base del que havíem implementat al nivell bàsic. Així que aquí només mencionarem els nous elements afegits respecte a l’anterior:

* Variables:
  + No hi ha cap canvi respecte al nivell bàsic.
* Predicats:
  + (no-dePersonal ?r - rover): indica que el rover r no transporta persones.
  + (no-deSuministro ?r - rover): indica que el rover r no transporta subministraments.
* Funcions:
  + No hi ha cap canvi respecte al nivell bàsic.
* Accions:
  + A base de les accions del nivell bàsic, en aquesta extensió només hem modificat les següents per tal de complir les restriccions definides a l’enunciat.
  + cargarPersonal(?r - rover ?a - asentamiento): carrega els personals de l’assentament a al rover r fins a tenir un màxim de 2 persones.
    - Precondicions: el rover r està buit o no transporta subministraments, el nombre de persones que transporta r és menor que 2, el rover r està a l’assentament a, l’assentament a és disponible.
  + revisaEstadoRover(?r - rover): actualitza el tipus de transport del rover r, o bé transporta personals o bé subministraments o bé està buit.
  1. **Problema**

En aquest apartat, amb el domini definit, el problema és idèntic que el de nivell bàsic.

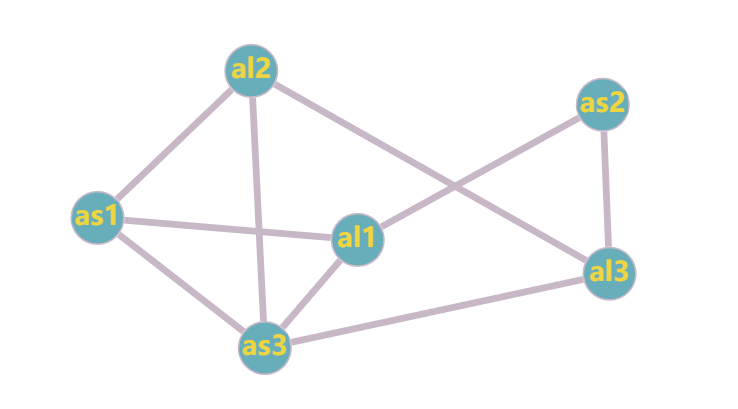
* 1. **Desenvolupament dels models**

El desenvolupament del problema d’aquesta extensió, hem dissenyat un prototip seguint l’extensió 1 guiada per l’enunciat de la pràctica. A més dels elements afegits a nivell bàsic, hem afegit nous elements del problema respecte a la restricció mencionada anteriorment.

* 1. **Jocs de prova**
* Joc de prova 1

Aquest joc de prova disposa d’un conjunt de 6 bases, 3 de tipus assentament i 3 de tipus magatzem, 2 rovers i 10 peticions.

Per ells, hem definit la connectivitat entre les bases d’aquesta manera:



On asX són els assentaments i alX són els magatzems.

Les disponibilitats dels magatzems són:

* al1: aquest magatzem disposa d’3 unitats de subministrament i 3 personals.
* al2: aquest magatzem disposa només de 2 unitats de subministraments.
* al3: aquest magatzem disposa d’1 unitat de subministrament i 1 personal.

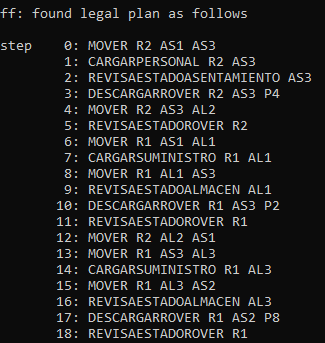
Per la part dels rovers, tots dos rovers r1 i r2 estan aparcats en l’assentament as1 i, òbviament, estan buits inicialment.

Per la part de les peticions, tenim que:

* p1: de tipus personal amb destinació d’as2.
* p2: de tipus subministrament amb destinació d’as3.
* p3: de tipus subministrament amb destinació d’as1.
* p4: de tipus personal amb destinació d’as3.
* p5: de tipus personal amb destinació d’as3.
* p6: de tipus subministrament amb destinació d’as1.
* p7: de tipus subministrament amb destinació d’as1.
* p8: de tipus subministrament amb destinació d’as2.
* p9: de tipus personal amb destinació d’as1.
* p10: de tipus personal amb destinació d’as3.

Amb aquesta inicialització esperem una planificació per tal de complir el nostre objectiu respectant la restricció proposada. La solució que genera serà vàlida, ja que el graf que connecta les bases resulta connex. Així que el rover pot arribar a qualsevol base des d’una base determinada.

En el següent mostrem la part del pla obtingut en l’output. El complet l’hem ajuntat en el fitxer [output1\_problema\_ext.txt] a part per poder veure amb més claredat.



* Joc de prova 2

1. **EXTENSIÓ 2**

Per aquesta extensió, a més de les restriccions mencionades a l’apartat anterior, s’afegeix que els rovers disposen d’un combustible limitat i carregat al principi del dia, de tal manera només poden realitzar un nombre de moviments limitats. En cada desplaçament entre bases es gasta una unitat de combustible. D’aquí disposem dues versions, una que no importa la quantitat total de combustible gastat en tots els rovers mentre que facin les entregues. L’altra apliquem una funció d’optimització per minimitzar el combustible total gastat.

* 1. **Domini**

Respecte a les restriccions dels apartats anteriors, hem afegit una nova restricció explicada anteriorment. Doncs, en aquest apartat només explicarem els nous afegits.

* Variables:
  + No hi ha cap canvi respecte a l’extensió 1.
* Predicats:
  + No hi ha cap canvi respecte a l’extensió 1.
* Funcions:
  + (combustible ?r - rover): la quantitat de combustible disponible del rover r en un dia.
  + (combustibleTotal): la suma dels combustibles de diferents rovers gastats en un dia (només afegim a la versió 2).
* Accions:
  + S'ha canviat la precondició i l’efecte de l’acció mover, per poder satisfer la nova restricció afegida.
  + mover(?r - rover ?b1 - base ?b2 - base): a més dels anteriors, disminuïm una unitat de combustible del rover r i incrementem una unitat de combustibleTotal (versió 2).
    - Precondicions: a més de les que ja teníem de l’extensió anterior, s’ha afegit que el combustible del rover r no pot ser buit.
  1. **Problema**

El problema d’aquesta extensió, amb el domini definit, el problema és una addicció de l’extensió 1. S’ha afegit les següents característiques:

* Estat inicial: a més de les inicialitzacions definides anteriorment, hem donat als rovers una quantitat de combustible i hem inicialitzat el variable combustibleTotal.
* Mètrica: hem definit que s’ha de minimitzar el combustibleTotal, que és el combustible total gastat pels rovers (només per la versió 2).

* 1. **Desenvolupament dels models**

El desenvolupament del problema d’aquesta extensió està basat en un disseny de prototip incremental, ja que es van ampliar a poc a poc les restriccions de cada extensió. Doncs, hem afegit nous principis respecte a la restricció d’aquesta extensió, que consisteix a donar un combustible limitat a cada rover.

* 1. **Jocs de prova**
* Joc de prova 1

Aquest joc de prova és més senzill que els anteriors perquè hem de tenir en compte que un joc de prova molt complicat pot tardar molt de temps en trobar una solució, ja que estem fent servir una recerca de BFS. Per tant, el joc disposa d’un conjunt de 5 bases, 3 de tipus assentament i 2 de tipus magatzem, 2 rovers i 5 peticions.

Per ells, hem definit la connectivitat entre les bases d’aquesta manera:

On asX són els assentaments i alX són els magatzems.

Les disponibilitats dels magatzems són:

* al1: aquest magatzem disposa d’1 unitat de subministrament.
* al2: aquest magatzem disposa només d’1 unitat de subministrament.
* as1: aquest assentament disposa d’1 personal.
* as2: aquest assentament no disposa cap personal.
* as3: aquest assentament disposa d’1 personal.

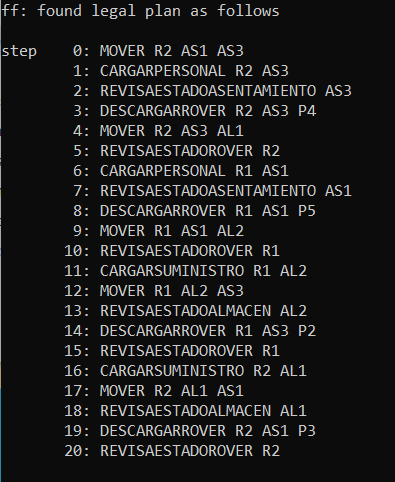
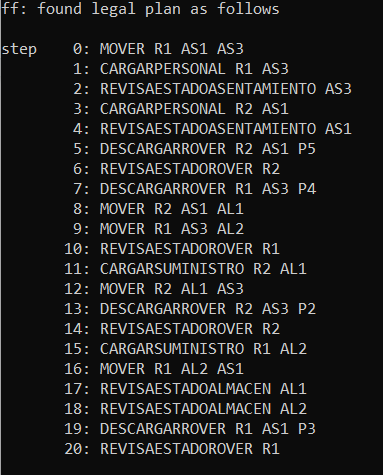
Per la part dels rovers, tots dos rovers r1 i r2 estan aparcats en l’assentament as1 i, òbviament, estan buits inicialment.

Per la part de les peticions, tenim que:

* p1: de tipus personal amb destinació d’as2.
* p2: de tipus subministrament amb destinació d’as3.
* p3: de tipus subministrament amb destinació d’as1.
* p4: de tipus personal amb destinació d’as3.
* p5: de tipus personal amb destinació d’as1.

Aquest estat inicial ens ajuda a obtenir una planificació per tal d’aconseguir el nostre objectiu respectant les restriccions proposades. La solució que genera serà vàlida, ja que el graf que connecta les bases resulta connex. Així que el rover pot arribar a qualsevol base des d’una base determinada i repartir els subministraments o personals de manera adequada.

En les dues següents imatges mostrem la part del pla obtingut en l’output de dues versions diferents, per tal de poder comparar-les. El complet l’hem ajuntat en els fitxers [output1\_problema\_ext2.1.txt] i [output1\_problema\_ext2.2.txt] a part per poder veure amb més claredat. Les solucions són com esperàvem, ja que els rovers han realitzat totes les peticions possibles.

La primera és de la versió 1, on no ha minimitzat el combustible total gastat dels rovers i la segona ho hem aplicat aquesta mètrica. Malgrat que observem que els desplaçaments són iguals en dues versions perquè es tracta d’un joc de prova molt simple, però el temps d’execució i l’estat de recerca són bastant diferents. Per tant, podem afirmar que la segona versió és l'òptima perquè s’ha intentat de fer una recerca de tots els estats possible per obtenir una solució més favorable respecte la mètrica que hem implementat.

* Joc de prova 2

1. **EXTENSIÓ 3**

A aquesta última extensió, addicionalment de les restriccions afegides a les anteriors extensions, es concreta que cada petició té una prioritat 1, 2 o 3 (sent 3 la màxima). De tal forma, busquem un pla que maximitzi la prioritat de les peticions servides.

* 1. **Domini**

Respecte a les restriccions dels apartats anteriors, hem afegit una nova restricció explicada anteriorment. Doncs, en aquest apartat només explicarem els nous afegits.

* Variables:
* Predicats:
* Funcions:
* Accions:
  1. **Problema**
  2. **Desenvolupament dels models**
  3. **Jocs de prova**

1. **CONCLUSIÓ**

En aquest treball hem resolt un problema simple i pràctic modalitzant-lo com un problema de planificació i utilitzant un sistema de planificació automàtica per la seva resolució. Malgrat que és un problema simple, ens ha permès ajudar-nos a entendre el tipus de problema del qual es pot aplicar amb les tècniques d’intel·ligència artificial apreses, concretament el llenguatge formal PDDL i l’eina *Fast Forward*.

En les diferents versions de la pràctica, hem analitzat el seu problema, que consisteix organitzar un repartiment de subministraments i personals als assentaments que han demanat les peticions apropiades, i hem modelitzat els elements que intervenen en aquest problema mitjançant objectes, predicats i accions de PDDL. Per ell, s’ha seguit un desenvolupament incremental del problema guiat per les extensions proposades en l’enunciat de la pràctica.

D’aquesta manera, hem pogut utilitzar el planificador Fast Forward per solucionar instàncies del problema escollides i estudiar els resultats obtinguts. Així mateix, hem pogut comprovar i entendre com els sistemes de planificació automàtics dissenyats de forma genèrica i sense coneixement del domini són capaços d’oferir solucions aproximades a la solució òptima en un temps d’execució molt raonable.