

Pràctica de Planificació

Projecte d'Intel·ligència Artificial

Grup 13

Joan Casahuga Altimiras

Miquel Torner Viñals

Xavier Bernat López



Departament de Ciències de la Computació

Universitat Politècnica de Catalunya

Octubre, 2022

Índex

1. Descripció del problema	2
1.1 Introducció	2
1.2 Anàlisi del problema	2
2. Adaptació del problema a PDDL	3
2.1 Nivell Bàsic	3
2.2 Extensió 1	6
2.2 Extensió 2	9
2.2 Extensió 3	11
3. Experimentació	13
3.1 Jocs de prova	13
3.2 Anàlisis de temps d'execució	18
4. Conclusions	20
5. Annex	21
5.1 Dades Influencia Persones	21
5.2 Dades Influencia Subministres	22
5.3 Dades Influencia Rovers	23
5.4 Dades Influencia Peticions	24

1. Descripció del problema

1.1 Introducció

L'objectiu d'aquesta pràctica és planificar el desplaçament de persones i subministraments a les destinacions de les seves peticions, utilitzant un planificador per construir la solució.

1.2 Anàlisi del problema

El problema planteja 6 elements diferenciats: rovers, persones, subministraments, bases, emmagatzematges i peticions:

- *Rovers*: permeten carregar i descarregar persones o subministraments en bases, amb la limitació de no poder portar persones i subministraments a la vegada a partir de la primera extensió.
- *Persones*: tenen la possibilitat de formar part d'una petició.
- *Subministrament*: tenen la possibilitat de formar part d'una petició.
- *Bases*: permeten la ubicació de persones.
- *Emmagatzematges*: permeten la ubicació de subministraments.
- *Peticions*: indiquen la destinació desitjada per una persona o subministrament.
- A partir d'aquesta informació adaptarem els elements i les accions a PDDL.

A cadascuna de les diferents extensions caldrà realitzar el transport de subministraments i persones afegint a cada extensió un element més a tenir en compte els quals veurem més endavant.

2. Adaptació del problema a PDDL

2.1 Nivell Bàsic

No existeixen limitacions sobre capacitats, l'objectiu únic és realitzar el màxim nombre de peticions possibles, aconseguint resoldre el problema en el moment en què totes les persones i subministraments hagin arribat al seu destí.

Per modelar el domini d'aquest nivell en PDDL s'han emprat els següents objectes (que també es faran servir en les extensions més endavant):

- rover (*object*): utilitzat per simular el rover com a tal.
- base (*object*): utilitzat per simular una base com a tal.
- asentamiento (*base*): utilitzat com a punt d'inici i fi de les persones.
- almacen (*base*): utilitzat com a punt d'inici i fi dels subministraments.
- carga (*object*): utilitzat per simular la càrrega que porta el rover.
- persona (*carga*): utilitzat per simular aquelles càrregues que representen persones.
- suministro (*carga*): utilitzar per simular aquelles càrregues que representen subministraments.

Pel que fa a predicats s'han usat els següents (els quals també s'utilitzaran per a les extensions de més endavant):

Predicat	Afirmació
esta_en ?c - carga ?b - base	Indica que la càrrega ?c està a la base ?b.
aparcado_en ?r - rover ?b - base	Indica que el rover ?r està a la base ?b.
esta_en_rover ?c - carga ?r - rover	Indica que la càrrega ?c està al rover ?r. És a dir que el rover porta aquesta càrrega.
hay_camino ?b1 - base ?b2 - base	Indica que hi ha un camí entre ?b1 i ?b2 pel qual el rover es pot moure.

Les accions que s'empren en aquest primer nivell són les següents:

mover_rover	
Paràmetres	
?r	Rover a moure.
?ori	Base d'origen del rover.
?des	Base destí del rover.
Precondició	
aparcado_en ?r ?ori	El rover a moure està aparcat a la base d'origen.
(hay_camino ?ori ?des) or (hay_camino ?des ?ori)	Hi ha un camí que porta de la base origen a la base destí o bé un que porta de la base destí a la base origen.
Efecte	
aparcado_en ?r ?des	El rover a moure es troba ara a la base destí.
not (aparcado_en ?r ?ori)	El rover a moure no es troba a la base d'origen.

subir_carga	
Paràmetres	
?r	Rover a on es vol col·locar la càrrega.
?c	Càrrega a col·locar al rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en ?c ?b	La càrrega per col·locar al rover està a la base b.
aparcado_en ?r ?b	El rover a carregar està aparcat a la base b.
Efecte	
not (esta_en ?c ?b)	La càrrega per col·locar al rover ja no es troba a la base b.
esta_en_rover ?c ?r	La càrrega està col·locada en el rover.

bajar_carga	
Paràmetres	
?r	Rover d'on es treu la càrrega.
?c	Càrrega que s'agafa del rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en_rover ?c ?r	La càrrega a descarregar està col·locada en el rover r.
aparcado_en ?r ?b	El rover a descarregar està aparcad a la base b.
Efecte	
not (esta_en_rover ?c ?r)	La càrrega ja no està col·locada en el rover.
esta_en ?c ?b	La càrrega a descarregar es troba a la base b.

Pel que fa a la definició del problema s'ha realitzat de la següent forma:

Objects:

Es declaren els objectes que componen el problema. En aquesta versió base, els objectes declarats són els diferents rovers, els assentaments, els magatzems, les persones i els subministraments.

Init:

Es declara l'estat inicial del problema. En aquest cas, tenim els diferents subministraments emmagatzemats als diferents magatzems (utilitzant el predicat *esta_en*), les persones ubicades als assentaments corresponents (tornant a utilitzar el predicat *esta_en*), els rovers aparcats en les bases escollides (en aquest cas, fent servir el predicat *aparcado_en*) i, finalment, els camins entre les diferents bases (siguin assentaments o magatzems), que es formen utilitzant el predicat *hay_camino*.

Goal:

Com a *Goal* definim l'objectiu del problema. En aquest cas, definim els llocs on volem que puguin acabar tant les persones com els subministraments declarats a *init*.

2.2 Extensió 1

S'incorpora la limitació que cada rover pot portar com a màxim un subministrament o dues persones (no es poden barrejar).

Pel que fa al modelatge del domini d'aquesta primera extensió s'han usat els mateixos objectes i predicats que en el nivell bàsic. Ara bé, donat que aquesta extensió cal limitar la capacitat de càrrega dels rovers hem decidit diferenciar si es carregava una persona o un subministrament, deixant l'acció de moure tal com està en el nivell base, però dividint la càrrega i la descàrrega en 2 accions diferents, en cadascun d'ells s'utilitza un fluent (*plazasLibres ?r - rover*) per incrementar o reduir el nombre de places lliures al rovers i així evitar que els rovers sobrepassin la seva capacitat límit.

subir_persona	
Paràmetres	
?r	Rover a on es vol col·locar la càrrega.
?p	Persona a col·locar al rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en ?p ?b	La persona que pujarà al rover està a la base b.
aparcado_en ?r ?b	El rover a carregar està aparcad a la base b.
(plazasLibres ?r) > 0	Hi ha com a mínim 1 espai al rover (el necessari per carregar una persona).
Efecte	
not (esta_en ?p ?b)	La persona a pujar al rover ja no es troba a la base b.
esta_en_rover ?p ?r	La persona està en el rover.
(plazasLibres ?r) -= 1	Es redueix en 1 el nombre de places disponibles.

subir_suministro	
Paràmetres	
?r	Rover a on es vol col·locar la càrrega.
?s	Subministrament a col·locar al rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en ?s ?b	El subministrament a col·locar al rover està a la base b.
aparcado_en ?r ?b	El rover a carregar està aparcad a la base b.
(plazasLibres ?r) > 1	Hi ha 2 espais al rover (el necessari per carregar un subministrament).
Efecte	
not (esta_en ?s ?b)	El subministrament a col·locar al rover ja no es troba a la base b.
esta_en_rover ?s ?r	El subministrament està col·locat en el rover.
(plazasLibres ?r) -= 2	Es redueix en 2 el nombre de places disponibles.

bajar_persona	
Paràmetres	
?r	Rover d'on es treu la càrrega.
?p	La persona que baixa del rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en_rover ?p ?r	La persona a baixar està en el rover r.
aparcado_en ?r ?b	El rover a descarregar està aparcad a la base b.
Efecte	
not (esta_en_rover ?p ?r)	La persona ja no està en el rover.
esta_en ?p ?b	La persona a baixar es troba a la base b.
(plazasLibres ?r) += 1	S'incrementa en 1 el nombre de places disponibles.

bajar_suministro	
Paràmetres	
?r	Rover d'on es treu la càrrega.
?s	Subministrament que s'agafa del rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en_rover ?s ?r	El subministrament a descarregar està col·locada en el rover r.
aparcado_en ?r ?b	El rover a descarregar està aparcad a la base b.
Efecte	
not (esta_en_rover ?s ?r)	El subministrament ja no està col·locat en el rover.
esta_en ?s ?b	El subministrament a descarregar es troba a la base b.
(plazasLibres ?r) += 2	S'incrementa en 2 el nombre de places disponibles.

Quant a la definició del problema, aquesta és la mateixa que en el nivell base, però s'afegeix un fluent fa que haguem de modificar només la funció de *init*. Ho fem afegint els valors inicials pel fluent *plazasLibres*, la qual ens permetrà incorporar la limitació de càrrega. Com que hem decidit que aquestes serviran tant per persones com per subministraments, les inicialitzem a 2 i ja les mateixes accions s'encarreguen de restar-li 1 o 2 a aquesta quantitat en funció de si el que puja és una persona o un subministrament.

2.2 Extensió 2

S'introdueix el combustible màxim, equivalent al nombre de moviments disponibles que el rover pot efectuar. La finalitat és comparar les execucions on es minimitza el combustible amb les que la quantitat de combustible gastat és indiferent.

En aquesta extensió pel que fa al modelatge de domini s'ha decidit afegir un fluent en els rovers el qual s'utilitza com a dipòsit de gasolina i així evitar que faci més moviments dels que se li permet (el qual s'inicialitza en el problema), també s'ha afegit un fluent global per evitar iterar per tots els rovers a l'hora de minimitzar. Quant a tipus i funcions no canvia, ara bé en les accions hi ha modificacions més concretament l'acció de moure.

mover_rover	
Paràmetres	
?r	Rover a moure.
?ori	Base d'origen del rover.
?des	Base destí del rover.
Precondició	
aparcado_en ?r ?ori	El rover a moure està aparcad a la base d'origen.
(hay_camino ?ori ?des) or (hay_camino ?des ?ori)	Hi ha un camí que porta de la base origen a la base destí o bé un que porta de la base destí a la base origen.
(combustible ?r) > 0	El rover té com a mínim una unitat de combustible.
Efecte	
aparcado_en ?r ?des	El rover a moure es troba ara a la base destí.
not (aparcado_en ?r ?ori)	El rover a moure no es troba a la base d'origen.
(combustible ?r) -= 1	Es redueix en una unitat el combustible del rover.
(combustibleTotal) += 1	S'augmenta en una unitat el combustible total utilitzat pels rovers.

Anàlogament, al que passava amb la primera extensió, com que al domini s'ha afegit un nou fluent, hem de determinar-ne un valor inicial a la funció *init* de la definició del problema. Inicialitzem, per tant, el combustible de cada rover amb la quantitat que vulguem i també inicialitzem el fluent *combustibleTotal* a 0, ja que en un inici cap rover està fent servir combustible, variable que servirà com a heurística per minimitzar la quantitat de combustible utilitzat en la segona versió.

Aquest *combustibleTotal* és el que, a la versió 2 d'aquesta extensió, es vol minimitzar, per tant, afegim a la definició del problema el (*:metric minimize (combustibleTotal)*).

2.2 Extensió 3

S'introdueix la prioritat de les peticions que poden ser de prioritat 1, 2 o 3, igual que en el combustible farem una versió on es maximitzi les prioritats (sense deixar de vista la minimització de combustible) i una en què no, per posteriorment fer la comparació.

En aquesta extensió pel que fa al modelatge de domini s'ha decidit afegir un fluent en les *càrregues* el qual s'utilitza com a prioritat, amb això li permet afegir a prioritats total amb l'objectiu de maximitzar la prioritat de les càrregues disposades segons les peticions, en el cas que volem tenir en compte el combustible amb les prioritats ajuntarem el recompte en un mateix valor que serà el total heurístic, que substituirà el combustibleTotal i prioritatsTotal per un de sol.

bajar_suministro	
Paràmetres	
?r	Rover d'on es treu la càrrega.
?s	Subministrament que s'agafa del rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en_rover ?s ?r	El subministrament a descarregar està col·locada en el rover r.
aparcado_en ?r ?b	El rover a descarregar està aparcad a la base b.
Efecte	
not (esta_en_rover ?s ?r)	El subministrament ja no està col·locat en el rover.
esta_en ?s ?b	El subministrament a descarregar es troba a la base b.
(plazasLibres ?r) += 2	S'incrementa en 2 el nombre de places disponibles.
(prioritatsTotal) += (prioritat ?s)	S'incrementa la variable proritatsTotal amb la del subministrament descarregat.

bajar_persona	
Paràmetres	
?r	Rover d'on es treu la càrrega.
?p	La persona que baixa del rover.
?b	Base on s'efectua l'operació.
Precondició	
esta_en_rover ?p ?r	La persona a baixar està en el rover r.
aparcado_en ?r ?b	El rover a descarregar està aparcad a la base b.
Efecte	
not (esta_en_rover ?p ?r)	La persona ja no està en el rover.
esta_en ?p ?b	La persona a baixar es troba a la base b.
(plazasLibres ?r) += 1	S'incrementa en 1 el nombre de places disponibles.
(prioritatsTotal) += (prioritat ?p)	S'incrementa la variable prioritatsTotal amb la de la persona descarregada.

En aquesta extensió, com que s'afegeix el fluent de *prioritatsTotal* i un de *prioritat* per a cada càrrega, tornem a modificar la funció *init* en aquest cas afegint un valor inicial per a *prioritatsTotal* de 0 (pel mateix motiu que el *combustibleTotal* de l'extensió anterior) i també inicialitzem la prioritat de cada una de les càrregues donant-li el valor que vulguem (de l'1 al 3).

També afegim al final de tot del problema (*:metric maximize (prioritatsTotal)*), per tal de maximitzar les prioritats i, per tant, fer que el planificador utilitzi abans els encàrrecs de prioritat alta, ja que maximitzen abans el fluent.

Per a la versió 2 de l'extensió, el que fem és unificar els fluents *prioritatsTotal* i *combustibleTotal* en un de sol, anomenat *TotalHeuristic*. L'únic que hem de modificar, doncs, és el (*:metric maximize (prioritatsTotal)*), que passa a ser (*:metric maximize (TotalHeuristic)*), i la funció *init*, ja que hem d'eliminar les inicialitzacions dels fluents *prioritatsTotal* i *combustibleTotal* i afegir la del *TotalHeuristic*.

3. Experimentació

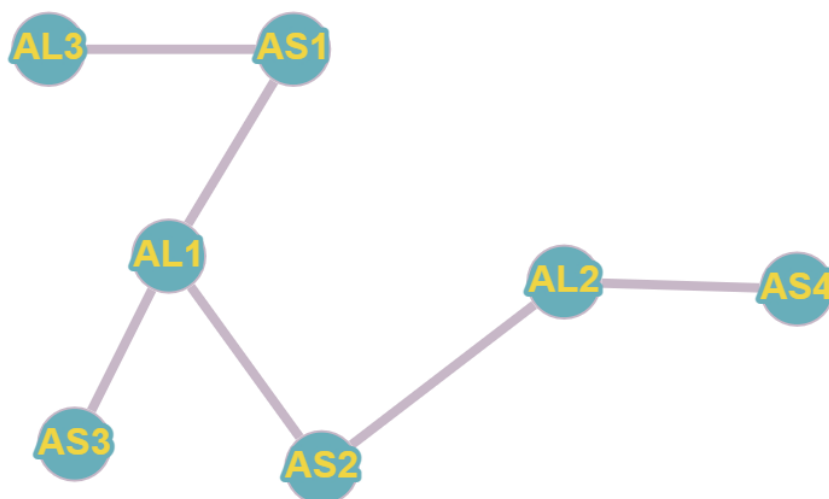
3.1 Jocs de prova

En aquest apartat veurem diferents jocs de prova no trivials i veurem com quina és la solució resultant de donar-los com a input en els nostres dominis.

Joc de prova 1:

L'objectiu d'aquest joc de proves és comprovar que l'extensió 1 funciona correctament, que no fa cap moviment il·legal i que arriba a la solució correctament.

El codi del problema es pot trobar a l'arxiu que té com a nom "test1.pddl". En aquest cas, el problema ens deixa el següent graf de magatzems i assentaments.



La persona 1 és a l'Assentament 1. Volem que acabi a l'assentament 4.

La persona 2 és a l'assentament 2. Volem que acabi a l'assentament 3.

La persona 3 és a l'assentament 2. Volem que acabi a l'assentament 4.

El subministrament 1 és al magatzem 1. Volem que acabi al magatzem 2.

El subministrament 2 és al magatzem 2. Volem que acabi al magatzem 3.

El subministrament 3 és al magatzem 1. Volem que acabi al magatzem 3.

El rover 1 és aparcat a l'assentament 1.

El rover 2 és aparcat a l'assentament 2.

El rover 3 és aparcat a l'assentament 3.

En executar-se, el planificador segueix les passes que es poden veure a test1.txt.

El primer que fa el planificador és anar a buscar el subministrament S1. Mou el rover R2 fins al AL1 i l'agafa. Després el mou fins l'AL2 (passant per l'AS2) i el deixa allà.

Després fa que el rover R1 vagi cap a l'assentament AS2 i reculli a la persona P2 per deixar-la a l'assentament AS3.

Mentrestant, el rover R3 es mou des de l'assentament AS3 fins al magatzem AL1 i agafa el subministrament S3. El porta al magatzem AL3 tot passant per l'assentament AS1.

El rover R2, que s'havia quedat al magatzem AL2, es mou cap a l'assentament AS2 i agafa a la persona P3. Tot seguit, es mou cap a l'assentament AS1 i la deixa allà. Un cop ha baixat a la persona P3, aprofita i puja la P1. El rover es recorre tot el camí fins arribar a l'assentament AS4, on deixa a P1.

El rover R3, que s'havia quedat al magatzem AL3, es mou fins arribar al magatzem AL2, on es carrega amb el subministrament S2 i fa tot el camí de tornada fins al AL3, on el deixa.

El subministrament S1 ha quedat al magatzem AL2 i, tant el S2 com el S3 han acabat al AL3, com volíem.

La persona P1 ha acabat al AS4, la persona P2 ha acabat al AS3 i P3 al AS1.

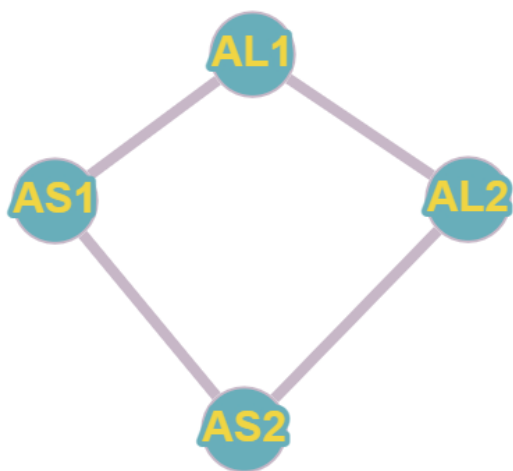
Podem observar, doncs, que l'extensió ha passat el joc de proves satisfactòriament, ja que, no ha fet cap moviment il·legal (anar d'un node a un altre quan no són connexos, pujar càrregues inexistents, baixar càrregues que no ha pujat prèviament, portar més càrrega de la permesa...) i ha arribat a l'estat solució.

Una observació a tenir en compte és que cap rover carrega dues persones alhora ni tampoc una combinació entre persones i subministraments.

Joc de prova 2:

L'objectiu d'aquest joc de proves és veure si l'extensió 2 funciona correctament i, a més, si la V2 minimitza el combustible. Per això el que fem és definir un problema amb un seguit de persones pensat perquè, si no pugen diverses persones alhora en un dels rovers, aquests es quedin gairebé sense combustible. El codi del problema es pot trobar als arxius que tenen com a nom "test2_v1.pddl" i "test2_v2.pddl".

El graf de les bases ens queda així:



La persona 1 és a l'Assentament 1. Volem que acabi a l'assentament 2.

La persona 2 és a l'assentament 1. Volem que acabi a l'assentament 2.

La persona 3 és a l'assentament 1. Volem que acabi a l'assentament 2.

La persona 4 és a l'assentament 1. Volem que acabi a l'assentament 2.

La persona 5 és a l'assentament 1. Volem que acabi a l'assentament 2.

La persona 6 és a l'assentament 2. Volem que acabi a l'assentament 1.

La persona 7 és a l'assentament 2. Volem que acabi a l'assentament 1.

La persona 8 és a l'assentament 2. Volem que acabi a l'assentament 1.

La persona 9 és a l'assentament 2. Volem que acabi a l'assentament 1.

El subministrament 1 és al magatzem 1. Volem que acabi al magatzem 2.

El subministrament 2 és al magatzem 1. Volem que acabi al magatzem 2.

El rover 1 és aparcat a l'assentament 1 i té 5 unitats de combustible.

El rover 2 és aparcat a l'assentament 2 i té 8 unitats de combustible.

L'output es pot veure en els arxius "test2v1.txt" i "test2v2.txt".

En aquest cas, el Rover 1 (que comença a AS1) comença movent-se cap al magatzem AL1 i movent la càrrega S1 fins al AL2. Tot seguit, es mou de nou al AL1 i desplaça la càrrega S2 al AL2. Allà acaba. En total, dels 5 desplaçaments que podia fer, n'ha fet 4.

El Rover 2 puja a P4 i a P5 i les mou a AS1 (recordem que R2 comença a AS2). Allà, els baixa i fa pujar a P1 i P2, i els mou cap a AS2. Torna cap a AS1 i puja a P3 per portar-la cap a AS2. En total, dels 8 desplaçaments que tenia disponibles, n'ha fet 4.

Podem observar que, efectivament, l'extensió 2 funciona bé, ja que, a més de no fer moviments il·legals (com ja passava amb l'extensió 1), respecta el combustible màxim de

cada Rover, sobretot en el cas del R2, que si no transportava a múltiples persones en cada viatge, no hauria tingut suficient combustible. Cal destacar que ara sí que el R2 ha fet que pugin diversos passatgers alhora.

En el cas de la versió 2 del problema (minimitzant combustible utilitzat), l'output no ha variat, ja que ja l'ha minimitzat en la V1. Així i tot, es pot veure l'output generat al fitxer "test2v2.txt"

Joc de prova 3:

En aquest joc de proves volem veure si l'extensió 3 funciona correctament i, a més, si la V2 minimitza el combustible alhora que maximitza la prioritat. Per fer-ho, tornarem a fer servir el graf del joc de proves anterior, però afegirem certes prioritats. El codi del problema es pot trobar als arxius que tenen com a nom "test3_v1.pddl" i "test3_v2.pddl".

La persona 1 és a l'Assentament 1 i té prioritat 2. Volem que acabi a l'assentament 2.

La persona 2 és a l'assentament 1 i té prioritat 2. Volem que acabi a l'assentament 2.

La persona 3 és a l'assentament 1 i té prioritat 1. Volem que acabi a l'assentament 2.

La persona 4 és a l'assentament 2 i té prioritat 1. Volem que acabi a l'assentament 1.

La persona 5 és a l'assentament 2 i té prioritat 2. Volem que acabi a l'assentament 1.

El subministrament 1 és al magatzem 1 i té prioritat 3. Volem que acabi al magatzem 2.

El subministrament 2 és al magatzem 1 i té prioritat 3. Volem que acabi al magatzem 2.

El rover 1 és aparcat a l'assentament 1 i té 5 unitats de combustible.

El rover 2 és aparcat a l'assentament 2 i té 8 unitats de combustible.

L'executem primer amb la versió 1 de l'extensió:

L'output es pot veure a l'arxiu "test3v1.txt".

El R1 comença movent-se de AS1 a AL1, i transportant el subministrament S1 cap a AL2.

Tot seguit, torna cap a AL1 i mou el subministrament S2 cap a AL2. Allà acaba. En total, ha fet 4 trajectes dels 5 que podia fer, i ha fet les dues tasques que eren de prioritat 3.

El R2 fa pujar tant a P4 com a P5, i els mou cap a AS1 (recordem que comença a AS2). Allà els fa baixar i puja a P1 i P2 (ambdós amb prioritat 2) i els mou cap a AS2. Torna a AS1 i s'emporta a P3 (amb prioritat 1) cap a AS2 de nou. En total, dels 8 viatges que podia fer, n'ha fet 4, i ha fet primer els que tenien una prioritat més alta (en el cas de P4 i P5 no hi havia cap altra persona allà, per tant, els ha fet alhora).

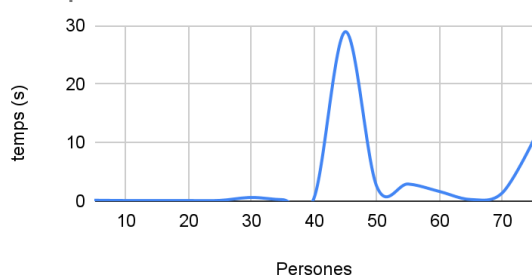
En el cas de la versió 2 de l'extensió, l'output és el mateix, ja que ha passat el que passava ja amb l'extensió 2, ja optimitza de per si el combustible utilitzat, almenys en aquesta resolució que ens dona. Tot i això, es pot veure l'output a l'arxiu "test3v2.txt".

3.2 Anàlisi de temps d'execució

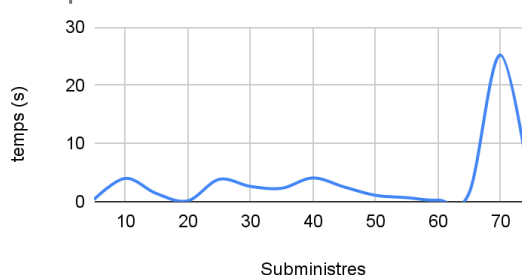
Per generar els múltiples tests hem creat un script amb python que permet decidir el nombre de tests, el temps de *timeout* dels tests, el tipus d'extensió i l'escala que indica l'increment de la mètrica entre tests, a partir d'aquestes hem generat amb escala 5, 15 tests (fent ús d'un segon script dedicat a la generació dels fitxers del problema en si) de l'extensió 2 i *timeout* de 5 minuts, amb les quals hem pogut observar quines eren les tendències:

Per les parts de persones i subministraments hem observat com el seu efecte no és massa important, tot i que pot existir pic en algunes de les execucions ho atribuïm a l'aleatorietat del generador.

Temps Execució / Persones

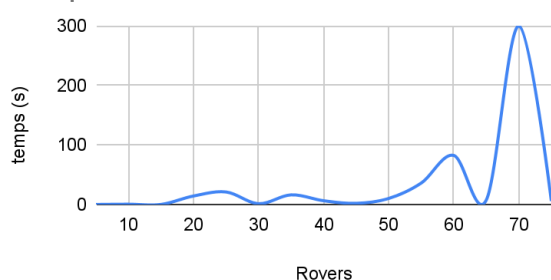


Temps Execució / Subministres

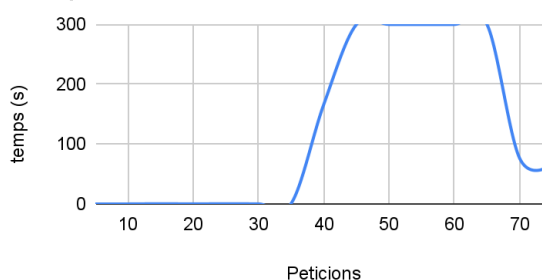


Per altra banda, tant els rovers com les peticions són una forma senzilla d'incrementar el temps d'execució, encara que d'una forma més gradual ho fan el número de rovers, en canvi, les peticions poden fer alterar el temps d'execució dràsticament.

Temps Execució / Rovers

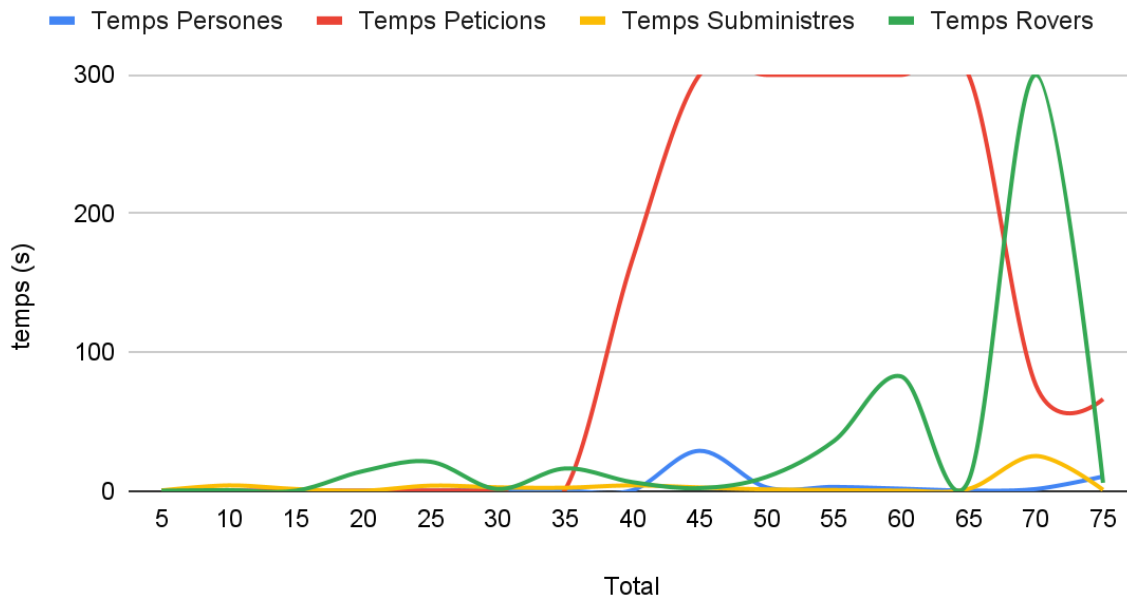


Temps Execució / Peticions



Per comparar les progressions amb les mateixes escales, hem solapat en aquest gràfic les execucions de persones, peticions, subministraments i rovers, on podem veure que els únics que sobrepassaven els 5 minuts d'execució han estat els rovers i les peticions, en canvi, les persones i subministraments, tot i que sí que tenen un petit efecte és negligible en visió de les altres dues contribucions.

Temps / Totals



4. Conclusions

Després de realitzar els experiments vistos en l'apartat anterior, ja podem donar les idees clau i les conclusions pel que fa a l'anàlisi que s'ha dut a terme per cada extensió, en les quals hem pogut observar com depenent de les condicions i mètriques utilitzades les solucions proposades s'ajustaven a l'objectiu marcat a més a més dels canvis en la forma de solucionar-ho.

La utilitat del programa dependrà de la mida de les mètriques, pensem que si tant el nombre de peticions com el número de rovers serà molt elevat i els moviments requereixen ser actualitzats al minut, segurament no és una opció viable i creiem que s'hauria de dividir en parts més petites per facilitar solucions menys bones però més ràpides.

A partir dels experiments realitzats en aquest treball, i de les conclusions extretes, hem arribat a la idea que seria útil realitzar més experiments sobre el temps d'execució amb escales més reduïdes i també explorar més casos.

Finalment, pensem que ha estat molt útil veure com resoldre un cas simple i interessant sobre com aplicar els sistemes de planificació, en el que hem pogut veure alguns dels seus fonaments i possiblement en el futur poder expandir sobre la manera d'enfocar i resoldre problemes.

5. Annex

Link del repositori utilitzat: <https://github.com/miquelt9/PDDL-MarsLogistics>

Link a les dades en cru:  Experimentació PDDL

Execució realitzada en base a l'extensió 2

5.1 Dades Influencia Persones

0.11465892399428412 test_persones5.pddl
0.21497052400081884 test_persones5.pddl
0.12273441399884177 test_persones10.pddl
0.11472039400541689 test_persones10.pddl
0.11436663500353461 test_persones15.pddl
0.11423396699683508 test_persones15.pddl
0.11432008299743757 test_persones20.pddl
0.1652533439992112 test_persones20.pddl
0.6161678990029031 test_persones25.pddl
0.6664704959985102 test_persones25.pddl
0.2647016750051989 test_persones30.pddl
0.26552757900208235 test_persones30.pddl
0.5154784279948217 test_persones35.pddl
0.5153954810011783 test_persones35.pddl
0.5158444150001742 test_persones40.pddl
0.5155957269962528 test_persones40.pddl
29.036576714002877 test_persones45.pddl
28.892990161999478 test_persones45.pddl
2.4205788819963345 test_persones50.pddl
2.6713358980050543 test_persones50.pddl
2.9807273799960967 test_persones55.pddl
2.901352612003393 test_persones55.pddl
1.81903552699805 test_persones60.pddl
1.5183572330060997 test_persones60.pddl
0.26489021200541174 test_persones65.pddl
0.31515975300135324 test_persones65.pddl
1.51814434100379 test_persones70.pddl
1.3177038479989278 test_persones70.pddl
11.045155605999753 test_persones75.pddl
9.887528103005025 test_persones75.pddl

5.2 Dades Influencia Subministres

0.6660380549947149 test_subministres5.pddl
0.11481386600644328 test_subministres5.pddl
4.175273547996767 test_subministres10.pddl
3.8735933670031955 test_subministres10.pddl
1.4679503179941094 test_subministres15.pddl
1.3682627659945865 test_subministres15.pddl
0.16496807400108082 test_subministres20.pddl
0.16477814000245417 test_subministres20.pddl
3.8745957459977944 test_subministres25.pddl
3.8237647370042396 test_subministres25.pddl
2.6210663099991507 test_subministres30.pddl
2.6706632999994326 test_subministres30.pddl
2.2201152940033353 test_subministres35.pddl
2.4194348770033685 test_subministres35.pddl
3.9884005580024677 test_subministres40.pddl
4.193385537000722 test_subministres40.pddl
2.5709896789994673 test_subministres45.pddl
2.521416103001684 test_subministres45.pddl
1.1671845210003085 test_subministres50.pddl
1.0668197900013183 test_subministres50.pddl
0.6666698280023411 test_subministres55.pddl
0.7672357100018417 test_subministres55.pddl
0.3153288370012888 test_subministres60.pddl
0.26510156999574974 test_subministres60.pddl
1.5190568269972573 test_subministres65.pddl
1.5190867720011738 test_subministres65.pddl
25.08478527099942 test_subministres70.pddl
25.368152208000538 test_subministres70.pddl
0.8665995269984705 test_subministres75.pddl
0.7658572999935132 test_subministres75.pddl

5.3 Dades Influencia Rovers

0.11471827499917708 test_rovers5.pddl
0.21491346900438657 test_rovers5.pddl
0.6164912329986691 test_rovers10.pddl
0.5658701749998727 test_rovers10.pddl
0.21487214700027835 test_rovers15.pddl
0.21511202399415197 test_rovers15.pddl
13.40285973400023 test_rovers20.pddl
15.30803329400078 test_rovers20.pddl
21.52144218199828 test_rovers25.pddl
20.67292426700442 test_rovers25.pddl
1.4681744330009678 test_rovers30.pddl
1.4180876450045616 test_rovers30.pddl
16.51101547700091 test_rovers35.pddl
15.968304497000645 test_rovers35.pddl
6.532575286997599 test_rovers40.pddl
6.281636644998798 test_rovers40.pddl
2.121065247003571 test_rovers45.pddl
1.9701469059946248 test_rovers45.pddl
10.298513416004425 test_rovers50.pddl
10.145637107998482 test_rovers50.pddl
36.73343950099661 test_rovers55.pddl
35.47713238900178 test_rovers55.pddl
84.48787694999919 test_rovers60.pddl
80.64128168399475 test_rovers60.pddl
7.636224464004044 test_rovers65.pddl
7.437699181005883 test_rovers65.pddl
Overflow de més de 300 segons test_rovers70.pddl
Overflow de més de 300 segons test_rovers70.pddl
6.084596370994404 test_rovers75.pddl
5.7332525289966725 test_rovers75.pddl

5.4 Dades Influencia Peticions

0.06431635499757249 test_peticions5.pddl
0.032336039999790955 test_peticions5.pddl
0.06422623599792132 test_peticions10.pddl
0.06428202000097372 test_peticions10.pddl
0.4155056899980991 test_peticions15.pddl
0.36527811999985715 test_peticions15.pddl
0.11451688900706358 test_peticions20.pddl
0.11480188700079452 test_peticions20.pddl
0.42368712600000435 test_peticions25.pddl
0.36548467600368895 test_peticions25.pddl
0.21485744700476062 test_peticions30.pddl
0.21488417199725518 test_peticions30.pddl
0.8162657050052076 test_peticions35.pddl
0.766164442000445 test_peticions35.pddl
170.96823257399956 test_peticions40.pddl
163.4349125430017 test_peticions40.pddl
Overflow de més de 300 segons test_peticions45.pddl
Overflow de més de 300 segons test_peticions45.pddl
Overflow de més de 300 segons test_peticions50.pddl
Overflow de més de 300 segons test_peticions50.pddl
Overflow de més de 300 segons test_peticions55.pddl
Overflow de més de 300 segons test_peticions55.pddl
Overflow de més de 300 segons test_peticions60.pddl
Overflow de més de 300 segons test_peticions60.pddl
Overflow de més de 300 segons test_peticions65.pddl
Overflow de més de 300 segons test_peticions65.pddl
81.63367315600044 test_peticions70.pddl
71.14589832799538 test_peticions70.pddl
69.92401667300146 test_peticions75.pddl
63.6583701549971 test_peticions75.pddl