Practica 3 de Planificació

[Inteligencia Artificial]
2021-2022

Albert Bertran Serrano
Guillermo López Esteve
Gerard Queralt Ferré

Index

Problema

Hem de dissenyar un sistema de reserves per un hotel, que sigui capaç d'assignar les peticions de reserva que arriben a les habitacions del nostre hotel. Tant les habitacions com les reserves poden tenir una capacitat de 1 a 4 ocupants.
Se'ns demana que utilitzant PDDL trobem la millor solució d'assignació de reserves a habitacions. Al llarg del problema haurem d'anar millorant la solució per tenir en compte més paràmetres, com pot ser l'orientació i optimitzar els resultats obtinguts.

Domini

Nivell Bàsic

Per representar el nostre sistema per fer reserves a hotels hem representat tres tipus necessaris:

- La habitació (habitacio)
- La reserva (reserva)
- El dia de la reserva (dia)

Les relacions entre aquests objectes són les següents:

- Una habitació està lliure en un dia concret
- Una reserva és per un dia o més

Per formalitzar aquestes relacions hem fet dos predicats per gestionar-les i hem afegit un tercer que mira si la reserva es correcta.

- (ocupada ?habitacio - habitacio ?dia - dia)
  Aquest predicat és true si l'habitació està ocupada en el dia indicat, fals altrament.
- (reservada ?reserva - reserva ?dia - dia)
  Aquest segon predicat és true si la reserva es per el dia indicat, fals altrament.
- (correcta ?reserva - reserva)
  Finalment, aquest últim predicat és true si la reserva esta satisfeta, fals altrament.

Per poder gestionar la viabilitat de les habitacions pel que fa a la capacitat d'aquestes utilitzem dues funcions, una per obtenir la mida de l'habitació i l'altre per obtenir les places requerides per la reserva.

- (capacitat_habitacio ?habitacio - habitacio)
- (capacitat_reserva ?reserva - reserva)

Per últim ens fa falta l'acció per fer la reserva. Aquesta acció es diu "reservar" i rep com a paràmetres una habitació i una reserva (?habitacio - habitacio ?reserva - reserva). Com a precondició primer comprova que la reserva no estigui satisfeta, després es comprova la capacitat de l'habitació: (>= (capacitat_habitacio ?habitacio) (capacitat_reserva ?reserva)) i finalment es comprova que l'habitació estigui lliure per tots els dies que dura la reserva.
Per acabar l'efecte que es produeix és el següent:

- La reserva queda satisfeta
- L'habitació estarà ocupada durant tots els dies que dura la reserva

Problema

Per modelar el problema definim alguns objectes de tipus habitacio, reserva i dia, cada un identificat pel nom del tipus i seguit d'un nombre que va augmentant si afegim més objectes com per exemple:

```
habitacio1 - habitacio
reserva1 - reserva
dia1 - dia
dia2 - dia
```

Per inicialitzar els elements alliberem les habitacions per tots els dies que consten en el problema, utilitzant el predicat (lliure ?habitacio - habitacio), inicialitzar les reserves per cada dia de durada que té i finalment assignar una capacitat a l'habitació i una capacitat a la reserva anàlogament. (= (capacitat_habitacio habitacio1) *numPersones*).

Per acabar definim el següent objectiu que s'encarregarà que una solució només sigui valida si totes les reserves s'han satisfet.

```
(:goal (forall (?reserva - reserva) (correcta ?reserva)))
```

Extensió 1

En aquesta extensió hem hagut de fer possible donar una solució parcial, és a dir, que una o més reserves no tenen perquè estan assignades. Per fer això, hem afegit la acció "marcar_no_reservada", aquesta acció marca com a procesada aquelles reserves que no es poden assignar, ha sigut necessària implementar-la per poder tenir un goal que permets una solució parcial però que tingues en compte totes les reserves.

A més hem afegit dos contadores un de reserves correctes i un de reserves pendents. Aquest contadors seran utilitzats com a mètrica per saber com adequada és la solució.

Per últim, canviat el goal del problema, ara el que busquem és que totes les reserves hagin sigut processades, i maximitza el número d'habitacions correctes

```
(:goal (forall (?reserva - reserva) (processada ?reserva)))
(:metric maximize (+ (reserves_correctes) 0))
```

Aquesta métrica, només calcula el nombre de reserves assignades a una habitació i no maximitza el nombre de dies que les habitacions estan ocupades.

Extensió 2

En aquesta extensió hem d'intentar que l'orientació de les habitacions satisfaci la preferència de la reserva, però prioritzant encara el nombre de reserves.

Per a fer això, hem afegit dos noves funcions, orientacio_habitacio i orientacio_reserva, i representem les orientacions amb nombres (Nord -> 0, Est -> 1, Sud -> 2 i Oest -> 3). En el domini, un cop s'ha acceptat una reserva, comprovem que estigui orientada correctament; de no ésser així, incrementem una mètrica anomenada reserves_mal_orientades.

Utilitzant Metric-FF com en l'extensió 1, en el problema intentem minimitzar les habitacions no assignades (a l'inrevés de l'extensió 1, on maximitzàvem les assignades, per simplificar la mètrica) sumades a les reserves mal orientades; dit d'una altra manera, volem tenir el mínim de reserves no ateses i reserves mal orientades. Com em dit, però, prioritzem que hi hagi menys reserves no ateses, així que aquesta mètrica la multipliquem per 1000 com a ponderació, ja que preferim que hi hagi més reserves assignades que menys però mal orientades.

```
(:goal (forall (?reserva - reserva) (processada ?reserva)))
(:metric minimize (+ (* 1000 (reserves_pendents)) (reserves_mal_orientades)))
```

Extensió 3

Per a aquesta part ens desfem de la problemàtica de l'orientació de les habitacions. Ens centrarem a minimitzar el nombre de places sobrants per reserva. És a dir que el sistema prioritzi assignar una habitació amb 5 places a una reserva realitzada per un grup de 3 abans que per una persona individual.

Per fer-ho respecte a l'extensió anterior hem afegit la funció "desperdici_places" que actua com a comptador en tot moment, i cada cop que s'efectua l'acció reservar, s'incrementa en el diferencial entre la capacitat de l'habitació i la capacitat de la reserva, és a dir, una reserva d'una habitació de 4 places a un grup de 2 comporta un desperdici de 2 places.

Després en la mètrica indiquem que volem minimitzar aquest valor, perquè es quedi amb la solució que tingui aquest valor més reduït, ponderem reserves_pendents, ja que com a l'extensió 2, preferim més reserves que menys reserves amb menys desperdici de places.

```
(:goal (forall (?reserva - reserva) (processada ?reserva)))
(:metric minimize (+ (* 1000 (reserves_pendents))  (desperdici_places))))
```

Extensió 4

En aquesta última extensió volem minimitzar el nombre diferent d'habitacions que utilitza l'hotel, és a dir, per un nombre de reserves assignar el menor nombre d'habitacions possibles. Per aconseguir-ho hem afegit respecte l'extensió anterior una funció i un predicat nous:

- (habitacions_diferents)
  Funció que acumula el nombre d'habitacions que s'utilitzen com a mínim una vegada.
- (utilitzada ?habitacio - habitacio)
  Predicat que és true si l'habitació es utilitzada almenys una vegada, fals altrament.

A l'acció de reservar hem afegit una comprovació addicional que comprova si una habitació no ha estat utilitzada encara, i en cas que així sigui incrementa el valor de la funció (habitacions_diferents).

Problema

En el problema hem hagut de modificar la mètrica per tenir en compte el valor d'habitacions diferents. Donat que donem més pes a minimitzar el nombre de diferents habitacions que no pas al desperdici de places cal indicar-ho al càlcul de la mètrica. Perquè tingui més pes hem optat per multiplicar el valor (habitacions_diferents) per 90 ja que és el màxim desperdici que pot generar una reserva (1 persona a una habitació de 4 multiplicat per 30 dies = 90 places desperdiciades).

```
(:goal (forall (?reserva - reserva) (processada ?reserva)))
(:metric minimize (+ (* 1000 (reserves_pendents)) (+ (* 90 (habitacions_diferents))
(desperdici_places))))
```

Generador de Problemes

Per dissenyar els jocs de prova hem dissenyat un programa amb python que crea jocs de prova amb els paràmetres desitjats.

Els paràmetres que acepta són:

- Per quina extensió és, per defecte serà el nivell bàsic.
- Número d'habitacions, per defecte són 10.
- Número de reserves, per defecte són 15.
- Nom del fitxer de output, per defecte es output.pddl.
- Nombre de dies, per defecte són 30.

Amb aquest paràmetres podem crear una gran quantitat de jocs de prova. El programa conta amb una petita explicació de quins paràmetres permet, es pot obtenir fàcilment amb la comanda *"./generadorsJocs.py --help"*.

Per exemple, la comanda per generar un joc de prova amb 20 habitacions, 25 reserves de 28 dies per l'extensió 3 és:

```
$ ./generadorJocs.py -o JocProva.pddl -nh 5 -r 6 -e 3 -d 5
```

Un contingut del fitxer, ja que algunes coses són aleatòries, és:

```
(define (problem ext3) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    habitacio1 - habitacio
    habitacio2 - habitacio
    habitacio3 - habitacio
    habitacio4 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    reserva3 - reserva
    reserva4 - reserva
    reserva5 - reserva
    dia0 - dia
    dia1 - dia
    dia2 - dia
    dia3 - dia
    dia4 - dia
  )
  (:init
    (reservada reserva0 dia3)
    (reservada reserva1 dia4)
    (reservada reserva2 dia3)
    (reservada reserva2 dia4)
    (reservada reserva3 dia1)
    (reservada reserva4 dia4)
    (reservada reserva5 dia2)
    (reservada reserva5 dia3)
    (reservada reserva5 dia4)
    (= (capacitat_habitacio habitacio0) 1)
    (= (capacitat_habitacio habitacio1) 3)
    (= (capacitat_habitacio habitacio2) 2)
    (= (capacitat_habitacio habitacio3) 4)
    (= (capacitat_habitacio habitacio4) 1)
    (= (capacitat_reserva reserva0) 2)
    (= (capacitat_reserva reserva1) 2)
    (= (capacitat_reserva reserva2) 2)
    (= (capacitat_reserva reserva3) 1)
    (= (capacitat_reserva reserva4) 1)
    (= (capacitat_reserva reserva5) 1)
    (= (reserves_pendents) 0)
    (= (reserves_correctes) 0)
    (= (desperdici_places) 0)
  )
  (:goal (forall (?reserva - reserva) (processada ?reserva)))
  (:metric minimize (+ (* 1000 (reserves_pendents)) (* 1 (desperdici_places))))
)
```

Jocs de prova

Nivell Bàsic

Joc 1

(problemaNivellBasic.pddl)

```
(define (problem basic) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    habitacio1 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    dia1 - dia
    dia2 - dia
  )
  (:init
    (reservada reserva0 dia1)
    (reservada reserva1 dia1)
    (reservada reserva2 dia2)
    (= (capacitat_habitacio habitacio0) 3)
    (= (capacitat_habitacio habitacio1) 2)
    (= (capacitat_reserva reserva0) 2)
    (= (capacitat_reserva reserva1) 1)
    (= (capacitat_reserva reserva2) 3)
  )

  (:goal (forall (?reserva - reserva) (correcta ?reserva)))
)
```

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO0 RESERVA2
        1: RESERVAR HABITACIO1 RESERVA1
        2: RESERVAR HABITACIO0 RESERVA0
```

En aquest primer joc podem observar com respectant les capacitats s'aconsegueix assignar totes les reserves.

Joc autogenerat

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO9 RESERVA14
        1: RESERVAR HABITACIO9 RESERVA13
        2: RESERVAR HABITACIO8 RESERVA12
        3: RESERVAR HABITACIO7 RESERVA11
        4: RESERVAR HABITACIO9 RESERVA10
        5: RESERVAR HABITACIO8 RESERVA9
        6: RESERVAR HABITACIO9 RESERVA8
        7: RESERVAR HABITACIO8 RESERVA7
        8: RESERVAR HABITACIO4 RESERVA6
        9: RESERVAR HABITACIO3 RESERVA5
       10: RESERVAR HABITACIO7 RESERVA4
       11: RESERVAR HABITACIO7 RESERVA3
       12: RESERVAR HABITACIO9 RESERVA2
       13: RESERVAR HABITACIO7 RESERVA1
       14: RESERVAR HABITACIO4 RESERVA0
```

Aquest és el joc autogenerat, utilitzant la comanda: "*./generadorJocs.py -o jg-ex0.pddl -e 0 -nh 10 -r 15*", el codi es pot trobar a l'arxiu *jg-ex1.pddl* situat a la carpeta src.

Extensió 1

Joc 1

(problemaExtensio-joc1.pddl)

```
(define (problem ext1) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    reserva3 - reserva
    dia1 - dia
    dia2 - dia
  )
  (:init
    (reservada reserva0 dia1)
    (reservada reserva1 dia1)
    (reservada reserva2 dia2)
    (reservada reserva3 dia2)
    (= (capacitat_habitacio habitacio0) 3)
    (= (capacitat_reserva reserva0) 1)
    (= (capacitat_reserva reserva1) 1)
    (= (capacitat_reserva reserva2) 1)
    (= (capacitat_reserva reserva3) 1)
    (= (reserves_pendents) 0)
    (= (reserves_correctes) 0)
  )

  (:goal (forall (?reserva - reserva) (processada ?reserva)))
  (:metric maximize (+ (reserves_correctes) 0))
)
```

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO0 RESERVA3
        1: MARCAR_NO_RESERVADA RESERVA2
        2: RESERVAR HABITACIO0 RESERVA1
        3: MARCAR_NO_RESERVADA RESERVA0
```

Com podem veure en aquest joc, hi ha dues reserves per cada dia però només una habitació. El programa correctament asigna una reserva a cada dia.

Joc 2

(problemaExtensio-joc2.pddl)

```
(define (problem ext1) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    dia1 - dia
    dia2 - dia
  )
  (:init
    (reservada reserva0 dia1)
    (reservada reserva0 dia2)
    (reservada reserva1 dia1)
    (reservada reserva2 dia2)
    (= (capacitat_habitacio habitacio0) 3)
    (= (capacitat_reserva reserva0) 1)
    (= (capacitat_reserva reserva1) 1)
    (= (capacitat_reserva reserva2) 1)
    (= (reserves_pendents) 0)
    (= (reserves_correctes) 0)
  )

  (:goal (forall (?reserva - reserva) (processada ?reserva)))
  (:metric maximize (+ (reserves_correctes) 0))
)
```

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO0 RESERVA2
        1: RESERVAR HABITACIO0 RESERVA1
        2: MARCAR_NO_RESERVADA RESERVA0
```

En aquest joc de proves, volem observar que en el cas de poder assignar més reserves, el domini correctament assigna dues reserves en lloc de una.

Joc autogenerat

(jg-ex1.pddl)

Aquest és el joc autogenerat, utilitzant la comanda: "*./generadorJocs.py -o jg-ex1.pddl -e 1 -nh 4 -r 24*", el codi es pot trobar a l'arxiu *jg-ex1.pddl* situat a la carpeta src.

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO3 RESERVA23
        1: RESERVAR HABITACIO2 RESERVA22
        2: RESERVAR HABITACIO3 RESERVA21
        3: RESERVAR HABITACIO3 RESERVA20
        4: RESERVAR HABITACIO3 RESERVA19
        5: RESERVAR HABITACIO1 RESERVA18
        6: RESERVAR HABITACIO2 RESERVA17
        7: RESERVAR HABITACIO1 RESERVA16
        8: RESERVAR HABITACIO0 RESERVA15
        9: RESERVAR HABITACIO2 RESERVA14
       10: RESERVAR HABITACIO1 RESERVA13
       11: RESERVAR HABITACIO2 RESERVA12
       12: MARCAR_NO_RESERVADA RESERVA11
       13: RESERVAR HABITACIO1 RESERVA10
       14: RESERVAR HABITACIO3 RESERVA9
       15: RESERVAR HABITACIO0 RESERVA8
       16: MARCAR_NO_RESERVADA RESERVA7
       17: MARCAR_NO_RESERVADA RESERVA6
       18: MARCAR_NO_RESERVADA RESERVA5
       19: MARCAR_NO_RESERVADA RESERVA4
       20: MARCAR_NO_RESERVADA RESERVA3
       21: RESERVAR HABITACIO2 RESERVA2
       22: MARCAR_NO_RESERVADA RESERVA1
       23: RESERVAR HABITACIO2 RESERVA0
```

.

Extensió 2

Joc Autogenerat 1:

Joc generat amb la comanda:
$ ./generadorJocs.py -o problemaExtensio2-joc1.pddl -nh 3 -r 5 -d 7 -e 2

```
(define (problem ext2) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    habitacio1 - habitacio
    habitacio2 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    reserva3 - reserva
    reserva4 - reserva
    dia0 - dia
    dia1 - dia
    dia2 - dia
    dia3 - dia
    dia4 - dia
    dia5 - dia
    dia6 - dia
  )
  (:init
    (reservada reserva0 dia4)
    (reservada reserva0 dia5)
    (reservada reserva0 dia6)
    (reservada reserva1 dia4)
    (reservada reserva1 dia5)
    (reservada reserva1 dia6)
    (reservada reserva2 dia3)
    (reservada reserva3 dia4)
    (reservada reserva3 dia5)
    (reservada reserva3 dia6)
    (reservada reserva4 dia5)
    (reservada reserva4 dia6)
    (= (capacitat_habitacio habitacio0) 3)
    (= (capacitat_habitacio habitacio1) 4)
    (= (capacitat_habitacio habitacio2) 2)
    (= (capacitat_reserva reserva0) 1)
    (= (capacitat_reserva reserva1) 1)
    (= (capacitat_reserva reserva2) 1)
    (= (capacitat_reserva reserva3) 2)
    (= (capacitat_reserva reserva4) 2)
    (= (orientacio_habitacio habitacio0) 3)
    (= (orientacio_habitacio habitacio1) 2)
    (= (orientacio_habitacio habitacio2) 0)
    (= (orientacio_reserva reserva0) 2)
    (= (orientacio_reserva reserva1) 3)
    (= (orientacio_reserva reserva2) 3)
    (= (orientacio_reserva reserva3) 3)
    (= (orientacio_reserva reserva4) 2)
    (= (reserves_pendents) 0)
    (= (reserves_correctes) 0)
    (= (reserves_mal_orientades) 0)
  )
 (:goal (forall (?reserva - reserva) (processada ?reserva)))
 (:metric minimize (+ (* 1000 (reserves_pendents)) (reserves_mal_orientades)))
)
```

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO2 RESERVA4
        1: RESERVAR HABITACIO0 RESERVA3
        2: RESERVAR HABITACIO0 RESERVA2
        3: MARCAR_NO_RESERVADA RESERVA1
        4: RESERVAR HABITACIO1 RESERVA0
```

Veiem que, com en l'extensió 1, maximitza el nombre de reserves, però a més a més intenta assignar les reserves seguint l'orientació: a l'habitació 0, orientada a l'oest, se li assignen les reserves 2 i 3, que prefereixen estar orientades a l'oest; a l'habitació 1 se li assigna la reserva 0, ja que coincideix la preferència d'orientació al sud; i la reserva 4 s'assigna a l'habitació 2, ja que tot i no coincidir les orientacions, suposa un major nombre de reserves totals.

Joc Autogenerat 2:

Joc generat amb la comanda:
./generadorJocs.py -o problemaExtensio2-joc2.pddl -nh 10 -r 4 -d 7 -e 2

```
(define (problem ext2) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    habitacio1 - habitacio
    habitacio2 - habitacio
    habitacio3 - habitacio
    habitacio4 - habitacio
    habitacio5 - habitacio
    habitacio6 - habitacio
    habitacio7 - habitacio
    habitacio8 - habitacio
    habitacio9 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    reserva3 - reserva
    dia0 - dia
    dia1 - dia
    dia2 - dia
    dia3 - dia
    dia4 - dia
    dia5 - dia
    dia6 - dia
  )
```

```
  (:init
    (reservada reserva0 dia6)
    (reservada reserva1 dia3)
    (reservada reserva1 dia4)
    (reservada reserva1 dia5)
    (reservada reserva1 dia6)
    (reservada reserva2 dia5)
    (reservada reserva2 dia6)
    (reservada reserva3 dia6)
    (= (capacitat_habitacio habitacio0) 1)
    (= (capacitat_habitacio habitacio1) 3)
    (= (capacitat_habitacio habitacio2) 2)
    (= (capacitat_habitacio habitacio3) 1)
    (= (capacitat_habitacio habitacio4) 4)
    (= (capacitat_habitacio habitacio5) 3)
    (= (capacitat_habitacio habitacio6) 3)
    (= (capacitat_habitacio habitacio7) 2)
    (= (capacitat_habitacio habitacio8) 2)
    (= (capacitat_habitacio habitacio9) 1)
    (= (capacitat_reserva reserva0) 1)
    (= (capacitat_reserva reserva1) 2)
    (= (capacitat_reserva reserva2) 2)
    (= (capacitat_reserva reserva3) 2)
    (= (orientacio_habitacio habitacio0) 2)
    (= (orientacio_habitacio habitacio1) 3)
    (= (orientacio_habitacio habitacio2) 3)
    (= (orientacio_habitacio habitacio3) 1)
    (= (orientacio_habitacio habitacio4) 0)
    (= (orientacio_habitacio habitacio5) 0)
    (= (orientacio_habitacio habitacio6) 2)
    (= (orientacio_habitacio habitacio7) 2)
    (= (orientacio_habitacio habitacio8) 2)
    (= (orientacio_habitacio habitacio9) 3)
    (= (orientacio_reserva reserva0) 0)
    (= (orientacio_reserva reserva1) 1)
    (= (orientacio_reserva reserva2) 2)
    (= (orientacio_reserva reserva3) 3)
    (= (reserves_pendents) 0)
    (= (reserves_correctes) 0)
    (= (reserves_mal_orientades) 0)
  )
 (:goal (forall (?reserva - reserva) (processada ?reserva)))
 (:metric minimize (+ (* 1000 (reserves_pendents)) (reserves_mal_orientades)))
)
```

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO5 RESERVA0
        1: RESERVAR HABITACIO2 RESERVA3
        2: RESERVAR HABITACIO8 RESERVA1
        3: RESERVAR HABITACIO7 RESERVA2
```

En aquest joc hem generat moltes més habitacions que reserves: en aquestes condicions (i, igual que al nivell bàsic, sempre que la reserva càpiga en l'habitació) sempre es satisfà la preferència d'orientació. Veiem que l'única habitació que satisfà la preferència d'orientació de la reserva 1, la 3, no té la mida necessària, i per tant no la podem assignar.

Extensió 3

Joc 1

(problemaExtensió3.pddl)

```
(define (problem ext3) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    habitacio1 - habitacio
    habitacio2 - habitacio
    habitacio3 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    reserva3 - reserva
    reserva4 - reserva
    dia1 - dia
    dia2 - dia
    dia3 - dia
  )
  (:init
    (reservada reserva0 dia1)
    (reservada reserva1 dia1)
    (reservada reserva2 dia2)
    (reservada reserva3 dia2)
    (reservada reserva4 dia3)
    (= (capacitat_habitacio habitacio0) 10)
    (= (capacitat_habitacio habitacio1) 5)
    (= (capacitat_habitacio habitacio2) 15)
    (= (capacitat_habitacio habitacio3) 3)
    (= (capacitat_reserva reserva0) 10)
    (= (capacitat_reserva reserva1) 15)
    (= (capacitat_reserva reserva2) 5)
    (= (capacitat_reserva reserva3) 11)
    (= (capacitat_reserva reserva4) 3)
    (= (desperdici_places) 0)
    (= (reserves_correctes) 0)
    (= (reserves_pendents) 0)
  )

  (:goal (forall (?reserva - reserva) (processada ?reserva)))
  (:metric minimize (+ (* 1000 (reserves_pendents)) (* 1 (desperdici_places))))
)
```

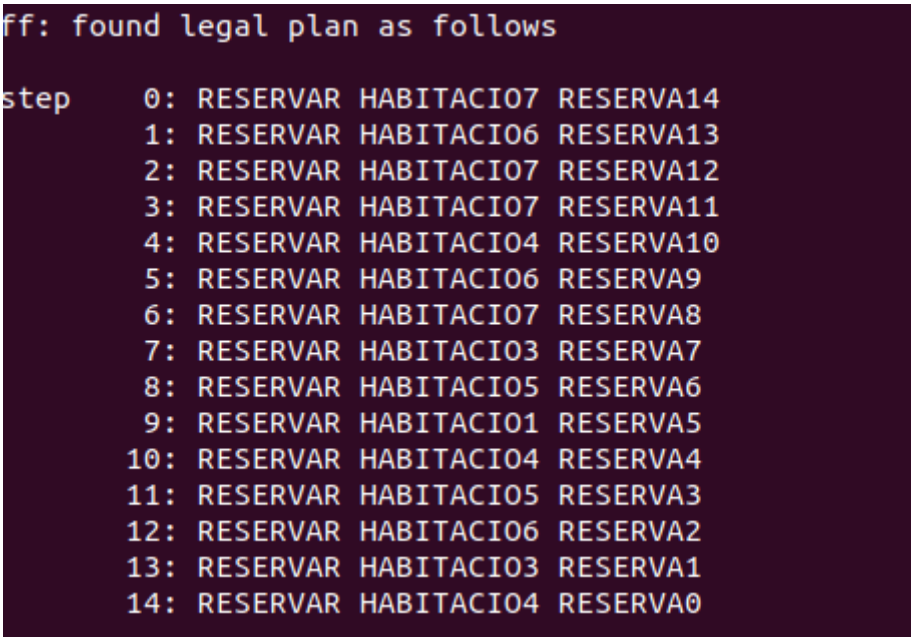
```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO2 RESERVA3
        1: RESERVAR HABITACIO2 RESERVA1
        2: RESERVAR HABITACIO3 RESERVA4
        3: RESERVAR HABITACIO1 RESERVA2
        4: RESERVAR HABITACIO0 RESERVA0
```

En el joc de proves veiem com per a cada habitació se li assigna la reserva amb la quantitat de places més semblant.

Joc autogenerat

(jg-ex3.pddl)

Aquest és el joc autogenerat, utilitzant la comanda: "*./generadorJocs.py -o jg-ex3.pddl -e 3 -nh 8 -r 15*", el codi es pot trobar a l'arxiu *jg-ex3.pddl* situat a la carpeta src.



Extensió 4

Joc 1

(problemaExtensió4.pddl)

```
(define (problem ext4) (:domain reserves)
  (:objects
    habitacio0 - habitacio
    habitacio1 - habitacio
    reserva0 - reserva
    reserva1 - reserva
    reserva2 - reserva
    dia1 - dia
    dia2 - dia
    dia3 - dia

  )
  (:init
    (reservada reserva0 dia1)
    (reservada reserva1 dia2)
    (reservada reserva2 dia3)
    (= (capacitat_habitacio habitacio0) 5)
    (= (capacitat_habitacio habitacio1) 4)
    (= (capacitat_reserva reserva0) 3)
    (= (capacitat_reserva reserva1) 2)
    (= (capacitat_reserva reserva2) 2)
    (= (desperdici_places) 0)
    (= (habitacions_diferents) 0)
    (= (reserves_correctes) 0)
    (= (reserves_pendents) 0)
  )

  (:goal (forall (?reserva - reserva) (processada ?reserva)))
  (:metric minimize (+ (* 1000 (reserves_pendents)) (+ (* 90 (habitacions_diferents)) (desperdici_places))))
)
```

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO1 RESERVA2
        1: RESERVAR HABITACIO1 RESERVA1
        2: RESERVAR HABITACIO1 RESERVA0
```

En aquest primer joc podem observar com escull una habitació, en aquest cas la 1 ja que es la que obté una millor métrica, ja que tot i que també es podrien assignar a l'habitació 0, aquesta última té un desperdici de places major.

Joc autogenerat:

Aquest és el joc autogenerat, utilitzant la comanda: "*./generadorJocs.py -o jg-ex4.pddl -e 4 -nh 3 -r 14*", el codi es pot trobar a l'arxiu *jg-ex4.pddl* situat a la carpeta src.

```
ff: found legal plan as follows

step    0: RESERVAR HABITACIO2 RESERVA13
        1: RESERVAR HABITACIO2 RESERVA12
        2: RESERVAR HABITACIO2 RESERVA11
        3: RESERVAR HABITACIO1 RESERVA10
        4: RESERVAR HABITACIO0 RESERVA9
        5: RESERVAR HABITACIO2 RESERVA8
        6: RESERVAR HABITACIO1 RESERVA7
        7: RESERVAR HABITACIO0 RESERVA6
        8: MARCAR_NO_RESERVADA RESERVA5
        9: RESERVAR HABITACIO0 RESERVA4
       10: MARCAR_NO_RESERVADA RESERVA3
       11: RESERVAR HABITACIO0 RESERVA2
       12: MARCAR_NO_RESERVADA RESERVA1
       13: MARCAR_NO_RESERVADA RESERVA0
```