

PRÁCTICA DE PLANIFICACIÓN: RESERVA DE HABITACIONES DE HOTELES

Grado en ingeniería informática
IA Curso 2021/22
Grupo 13

Roberto Amat Alins
Marc Camarillas Parés
Alejandro Salvat Navarro

Profesor: Javier Béjar

Índice

| | |
|-------------------------------------|-----------|
| 1. Modelado del dominio | 4 |
| 1.1 Nivel básico | 4 |
| 1.2 Extensión 1 | 5 |
| 1.3 Extensión 2 | 6 |
| 1.4 Extensión 3 | 8 |
| 1.5 Extensión 4 | 9 |
| 2. Modelado de los problemas | 11 |
| 2.1 Extensión 1 | 12 |
| 2.2 Extensión 2 | 12 |
| 2.3 Extensión 3 | 13 |
| 2.4 Extensión 4 | 14 |
| 3. Juegos de pruebas | 15 |
| 3.1 Nivel Básico | 15 |
| 3.1.1 Prueba 1 | 15 |
| 3.2 Extensión 1 | 15 |
| 3.2.1 Prueba 1 | 15 |
| 3.3 Extensión 2 | 16 |
| 3.3.1 Prueba 1 | 16 |
| 3.4 Extensión 3 | 16 |
| 3.4.1 Prueba 1 | 16 |
| 3.5 Extensión 4 | 17 |
| 3.5.1 Prueba 1 | 17 |
| 3.5.2 Prueba 2 | 17 |
| 4. Desarrollo de los modelos | 18 |
| 5. Experimentación | 18 |
| 6. Conclusiones | 20 |
| A. Anexo de Juegos De Prueba | 21 |
| A.1 Juego de Pruebas Nivel Básico | 21 |
| A.1.1 Juego de prueba 1 | 21 |
| A.1.1.1 Problema | 21 |
| A.1.1.2 Solución | 22 |
| A.1.2 Juego de prueba 2 | 22 |
| A.1.2.1 Problema | 22 |
| A1.2.2 Solución | 23 |

| | |
|------------------------------------|-----------|
| A.2 Juego de Pruebas Extensión 1 | 25 |
| A.2.1 Juego de prueba 1 | 25 |
| A.2.1.1 Problema | 25 |
| A.2.1.2 Solución | 26 |
| A.2.2 Juego de prueba 2 | 26 |
| A.2.2.1 Problema | 26 |
| A.2.2.2 Solución | 27 |
| A.3 Juego de Pruebas Extensión 2 | 29 |
| A.3.1 Juego de prueba 1 | 29 |
| A.3.1.1 Problema | 29 |
| A.3.1.2 Solución | 30 |
| A.3.2 Juego de prueba 2 | 31 |
| A.3.2.1 Problema | 31 |
| A.3.2.2 Solución | 32 |
| A.4 Juego de Pruebas Extensión 3 | 33 |
| A.4.1 Juego de prueba 1 | 33 |
| A.4.1.1 Problema | 33 |
| A.4.1.2 Solución | 34 |
| A.4.2 Juego de prueba 2 | 34 |
| A.4.2.1 Problema | 34 |
| A.4.2.2 Solución | 35 |
| A.5 Juego de Pruebas Extensión 4 | 37 |
| A.5.1 Juego de prueba 1 | 37 |
| A.5.1.1 Problema | 37 |
| A.5.1.2 Solución | 38 |
| A.5.2 Juego de prueba 2 | 38 |
| A.5.2.1 Problema | 38 |
| A.5.2.2 Solución | 39 |
| B. Anexo de Experimentación | 40 |
| B.1 (1 habitación 5 reservas) | 40 |
| B.1.1 Problema | 40 |
| B.1.2 Solución | 41 |
| B.2 (2 habitaciones 10 reserva) | 42 |
| B.2.1 Problema | 42 |
| B.2.2 Solución | 43 |
| B.3 (3 habitaciones 15 reserva) | 44 |
| B.3.2 Solución | 45 |
| B.4 (4 habitaciones 20 reserva) | 46 |

| | |
|---------------------------------|----|
| B.4.2 Solución | 48 |
| B.5 (5 habitaciones 25 reserva) | 49 |
| B.5.1 Problema | 49 |
| B.5.2 Solución | 51 |
| B.6 (6 habitaciones 30 reserva) | 52 |
| B.6.1 Problema | 52 |
| B.6.2 Solución | 55 |
| B.7 (7 habitaciones 35 reserva) | 56 |
| B.7.1 Problema | 56 |
| B.7.2 Solución | 60 |

1. Modelado del dominio

1.1 Nivel básico

Variables:

Los objetos que necesitaremos para modelar el dominio del nivel básico son:

- **Habitación** → Indica una habitación del hotel.
- **Reserva** → Indica una petición de reserva en el hotel, que se asignará o no dependiendo de la disponibilidad.

Predicados:

Los predicados que usaremos para resolver el problema son los siguientes:

- **(reservada ?res - reserva ?hab - habitacion)** → Indica si una habitación está asignada por una reserva concreta.
- **(resuelta ?res - reserva)** → Indica si una reserva ya ha sido asignada a una habitación.

Functions:

Las funciones que usaremos para resolver el problema son los siguientes:

- **(plazas ?hab - habitacion)** → Cantidad de gente que cabe en una habitación.
- **(personas ?res - reserva)** → Cantidad de gente que forma parte de una reserva.
- **(inicio ?res - reserva)** → Inicio de una reserva en una habitación.
- **(fin ?res - reserva)** → Final de una reserva en una habitación.

Acciones:

Las acciones que usaremos para resolver el problema son los siguientes:

- **asignar_habitacion** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación y que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva.
 - efecto → reservar habitación con la reserva y resolver reserva

1.2 Extensión 1

Variables:

Los objetos que necesitaremos para modelar el dominio del nivel básico son:

- **Habitación** → Indica una habitación del hotel.
- **Reserva** → Indica una petición de reserva en el hotel, que se asignará o no dependiendo de la disponibilidad.

Predicados:

Los predicados que usaremos para resolver el problema son los siguientes:

- **(reservada ?res - reserva ?hab - habitacion)** → Indica si una habitación está asignada por una reserva concreta.
- **(resuelta ?res - reserva)** → Indica si una reserva ya ha sido asignada a una habitación.

Functions:

Las funciones que usaremos para resolver el problema son los siguientes:

- **(reservasNoAsignadas)** → Número de reservas no asignadas.
- **(plazas ?hab - habitacion)** → Cantidad de gente que cabe en una habitación.
- **(personas ?res - reserva)** → Cantidad de gente que forma parte de una reserva.
- **(inicio ?res - reserva)** → Inicio de una reserva en una habitación.
- **(fin ?res - reserva)** → Final de una reserva en una habitación.

Acciones:

Las acciones que usaremos para resolver el problema son los siguientes:

- **reservar_Habitacion** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación y que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva.
 - efecto → reservar habitación con la reserva y resolver reserva

- **denegar_Reserva** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta y que el número de plazas reservadas sea mayor al número de plazas que ofrece la habitación o que alguna otra reserva en esa habitación durante el periodo de tiempo de la reserva.
 - efecto → marcar reserva como resuelta y aumentar en 1 el número de reservas no asignadas

1.3 Extensión 2

Variables:

Los objetos que necesitaremos para modelar el dominio del nivel básico son:

- **Habitación** → Indica una habitación del hotel.
- **Reserva** → Indica una petición de reserva en el hotel, que se asignará o no dependiendo de la disponibilidad.

Predicados:

Los predicados que usaremos para resolver el problema son los siguientes:

- **(reservada ?res - reserva ?hab - habitacion)** → Indica si una habitación está asignada por una reserva concreta.
- **(resuelta ?res - reserva)** → Indica si una reserva ya ha sido asignada a una habitación.

Functions:

Las funciones que usaremos para resolver el problema son los siguientes:

- **(total-cost)** → Penalización que queremos minimizar.
- **(reservasNoAsignadas)** → Número de reservas no asignadas.
- **(plazas ?hab - habitacion)** → Cantidad de gente que cabe en una habitación.
- **(pcardinalhab ?hab - habitacion)** → Punto cardinal al que esta orientada la habitación.
- **(pcardinalres ?res - reserva)** → Punto cardinal al que quiere estar orientada la reserva.
- **(personas ?res - reserva)** → Cantidad de gente que forma parte de una reserva.

- **(inicio ?res - reserva)** → Inicio de una reserva en una habitación.
- **(fin ?res - reserva)** → Final de una reserva en una habitación.
- **(num-res)** → Número de reservas

Acciones:

Las acciones que usaremos para resolver el problema son los siguientes:

- **reservar_Habitacion_Perfecta** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación, que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva y que el punto cardinal al que está orientada la habitación sea la preferente por la reserva.
 - efecto → reservar habitación con la reserva y resolver reserva
- **reservar_Habitacion** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación y que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva.
 - efecto → reservar habitación con la reserva, resolver reserva e incrementar en 1 el coste total de la solución
- **denegar_Reserva** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta y que el número de plazas reservadas sea mayor al número de plazas que ofrece la habitación o que alguna otra reserva en esa habitación durante el periodo de tiempo de la reserva.
 - efecto → marcar reserva como resuelta y aumentar en el coste total, tanto como el número de reservas que existen en el problema

1.4 Extensión 3

Variables:

Los objetos que necesitaremos para modelar el dominio del nivel básico son:

- **Habitación** → Indica una habitación del hotel.
- **Reserva** → Indica una petición de reserva en el hotel, que se asignará o no dependiendo de la disponibilidad.

Predicados:

Los predicados que usaremos para resolver el problema son los siguientes:

- **(reservada ?res - reserva ?hab - habitacion)** → Indica si una habitación está asignada por una reserva concreta.
- **(resuelta ?res - reserva)** → Indica si una reserva ya ha sido asignada a una habitación.

Functions:

Las funciones que usaremos para resolver el problema son los siguientes:

- **(reservasNoAsignadas)** → Número de reservas no asignadas.
- **(plazasMalgastadas)** → Número total de plazas de habitaciones sin ocupar.
- **(plazas ?hab - habitacion)** → Cantidad de gente que cabe en una habitación.
- **(personas ?res - reserva)** → Cantidad de gente que forma parte de una reserva.
- **(inicio ?res - reserva)** → Inicio de una reserva en una habitación.
- **(fin ?res - reserva)** → Final de una reserva en una habitación.

Acciones:

Las acciones que usaremos para resolver el problema son los siguientes:

- **reservar_Habitacion** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación y que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva.
 - efecto → reservar habitación con la reserva, resolver reserva e incrementar el número de plazas malgastadas en: plazas de la habitación - número de personas de la reserva

- **denegar_Reserva** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta y que el número de plazas reservadas sea mayor al número de plazas que ofrece la habitación o que alguna otra reserva en esa habitación durante el periodo de tiempo de la reserva.
 - efecto → marcar reserva como resuelta y aumentar en 1 el número de reservas no asignadas

1.5 Extensión 4

Variables:

Los objetos que necesitaremos para modelar el dominio del nivel básico son:

- **Habitación** → Indica una habitación del hotel.
- **Reserva** → Indica una petición de reserva en el hotel, que se asignará o no dependiendo de la disponibilidad.

Predicados:

Los predicados que usaremos para resolver el problema son los siguientes:

- **(reservada ?res - reserva ?hab - habitacion)** → Indica si una habitación está asignada por una reserva concreta.
- **(resuelta ?res - reserva)** → Indica si una reserva ya ha sido asignada a una habitación.
- **(usada ?hab - habitacion)** → Indica si una habitación ha sido usada para alguna reserva

Functions:

Las funciones que usaremos para resolver el problema son los siguientes:

- **(reservasNoAsignadas)** → Número de reservas no asignadas.
- **(plazasMalgastadas)** → Número total de plazas de habitaciones sin ocupar.
- **(plazas ?hab - habitacion)** → Cantidad de gente que cabe en una habitación.
- **(personas ?res - reserva)** → Cantidad de gente que forma parte de una reserva.

- **(inicio ?res - reserva)** → Inicio de una reserva en una habitación.
- **(fin ?res - reserva)** → Final de una reserva en una habitación.
- **(habitacionesDiferentes)** → Número total de habitaciones diferentes usadas para asignar reservas

Acciones:

Las acciones que usaremos para resolver el problema son los siguientes:

- **reservar_Habitacion** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación y que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva.
 - efecto → reservar habitación con la reserva, resolver reserva e incrementar el número de plazas malgastadas en: plazas de la habitación - número de personas de la reserva
- **denegar_Reserva** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta y que el número de plazas reservadas sea mayor al número de plazas que ofrece la habitación o que alguna otra reserva en esa habitación durante el periodo de tiempo de la reserva.
 - efecto → marcar reserva como resuelta y aumentar en 1 el número de reservas no asignadas
- **reservar_Habitacion_no_Usada** ⇒
 - parámetros → una reserva y una habitación
 - precondition → que la reserva no esté resuelta, que el número de plazas reservadas sea menor al número de plazas que ofrece la habitación, que no haya otras reservas a esa habitación en el periodo de tiempo de la reserva y que la habitación no haya sido usada para ninguna otra reserva.
 - efecto → reservar habitación con la reserva, resolver reserva, incrementar el número de plazas malgastadas en: plazas de la habitación - número de

personas de la reserva e incrementa en 1 el número de habitaciones diferentes usadas para satisfacer las reservas.

2. Modelado de los problemas

Todos los problemas tienen dos tipos de objetos que hay que asignar:

- reservas (r)
- habitaciones (h)

Para cada habitación o reserva que se quiera crear, primero hay que declarar el objeto en objects. Además hemos hecho que el generador de problemas te cree tantas habitaciones y reservas como le pidas.

Además todos los problemas tienen en común las siguientes funciones o predicados:

- reservada → inicialmente la asignamos a falso.
- plazas → se asigna un número entre 1 y 4.
- personas → se asigna un número entre 1 y 4.
- días → se asigna un número entre 1 y 29.
- fin → se asigna un número entre 2 y 30.

El generador de problemas ya está preparado para que los números dentro de este rango. Esto conformaría el estado inicial del nivel básico y en los siguientes apartados explicaremos lo que hemos tenido que añadirle para que se adecue a la extensión.

Para el estado final, todas las extensiones tienen la misma definición, pero cada una tiene una heurística diferente para optimizar la solución. (En el nivel básico no se optimiza nada).

Con todo esto ya tendríamos definidos los objetos que usaremos, el estado inicial y el estado final que queremos lograr.

2.1 Extensión 1

Además de lo básico, hemos añadido:

Solución inicial:

- `reservasNoAsignadas` → inicialmente lo asignamos siempre a 0.

Heurística:

$$h = \text{ReservasNoAsignadas}$$

- Queremos minimizar h .
- Para cada reserva que queda sin asignar, se aumenta en uno el número de reservas no asignadas

2.2 Extensión 2

Además de lo básico hemos añadido:

Solución inicial:

- `total-cost` → inicialmente lo asignamos siempre a 0.
- `pcardinalres` → obtiene valores entre 0 y 3 (N/S/E/O).
- `pcardinalhab` → obtiene valores entre 0 y 3 (N/S/E/O).

Heurística:

$$h = \text{totalCost}$$

- Queremos minimizar h .
- Hemos definido `totalCost` como una penalización que se impone a las habitaciones no reservadas y a las que no tienen orientación perfecta. Para maximizar el número de habitaciones asignadas, penalizamos con:
 - un punto si la habitación no tiene la orientación pedida por la reserva.
 - "`numReservas`" puntos si la reserva no se asigna a ninguna habitación.

2.3 Extensión 3

Además de lo básico hemos añadido:

Solución inicial:

- `reservasNoAsignadas` → inicialmente lo asignamos siempre a 0.
- `plazasMalgastadas` → inicialmente lo asignamos siempre a 0.

Heurística:

$$h = \text{ReservasNoAsignadas} \cdot \alpha + \text{PlazasMalgastadas}$$

- Queremos minimizar h .
- Para hacer que se minimicen las reservas no asignadas antes que los otros términos hemos usado un ponderador (α) para que penalice más.
 - $\alpha = 3 \cdot \text{NumReservas}$
- Hemos establecido α así debido a que así la penalización de plazas malgastadas nunca será mayor al de reservas no asignadas.

2.4 Extensión 4

Además de lo básico hemos añadido:

Solución inicial:

- `reservasNoAsignadas` → inicialmente lo asignamos siempre a 0.
- `plazasMalgastadas` → inicialmente lo asignamos siempre a 0.
- `habitacionesDiferentes` → inicialmente lo asignamos siempre a 0.

Heurística:

$$h = \text{ReservasNoAsignadas} \cdot \alpha + \text{PlazasMalgastadas} + \text{HabitacionesDiferentes} \cdot \beta$$

- Queremos minimizar h .
- Para hacer que se minimicen las reservas no asignadas antes que los otros términos hemos usado un ponderador (α) para que penalice más. A su vez hemos hecho lo mismo para minimizar el número de habitaciones diferentes antes que las plazas malgastadas, en este caso hemos usado el ponderador (β).
 - $\beta = 3 \cdot \text{NumReservas}$
- Hemos establecido β así debido a que así la penalización de plazas malgastadas nunca será mayor al de habitaciones diferentes.
 - $\alpha = \beta \cdot n + 3 \cdot \text{NumReservas}$
- Hemos establecido α así debido a que así la penalización de plazas malgastadas más las habitaciones diferentes nunca será mayor al de reservas no asignadas.

3. Juegos de pruebas

3.1 Nivel Básico

3.1.1 Prueba 1

Entrada:

Problema generado con el generador aleatorio (2 habitaciones, 6 reservas).
(Problema en [A.1.1] de los Anexos)

Solución Esperada:

No creemos que se satisfazcan todas las reservas debido a que al ser generada al azar muchos días se solapan. Por lo tanto no se asignará ninguna.

Solución Obtenida:

Como hemos explicado anteriormente no se han satisfecho todas las reservas debido a que algunas se solapan.
(Solución en [A.1.2] de los Anexos)

3.2 Extensión 1

3.2.1 Prueba 1

Entrada:

Problema generado con el generador aleatorio (2 habitaciones, 6 reservas).
(Problema en [A.2.1] de los Anexos)

Solución Esperada:

No creemos que se satisfazcan muchas reservas debido a que al ser generada al azar muchos días se solapan.

Solución Obtenida:

Como hemos explicado anteriormente no se han satisfecho todas las reservas debido a que algunas se solapan.
(Solución en [A.2.2] de los Anexos)

3.3 Extensión 2

3.3.1 Prueba 1

Entrada:

Problema generado con el generador aleatorio (2 habitaciones, 6 reservas).
(Problema en [A.3.1] de los Anexos)

Solución Esperada:

No creemos que se satisfazcan muchas reservas debido a que al ser generada al azar muchos días se solapan.

Solución Obtenida:

Como hemos explicado anteriormente no se han satisfecho todas las reservas debido a que algunas se solapan.
(Solución en [A.3.2] de los Anexos)

3.4 Extensión 3

3.4.1 Prueba 1

Entrada:

Problema generado con el generador aleatorio (2 habitaciones, 6 reservas).
(Problema en [A.4.1] de los Anexos)

Solución Esperada:

No creemos que se satisfazcan muchas reservas debido a que al ser generada al azar muchos días se solapan.

Solución Obtenida:

Como hemos explicado anteriormente no se han satisfecho todas las reservas debido a que algunas se solapan.
(Solución en [A.4.2] de los Anexos)

3.5 Extensión 4

3.5.1 Prueba 1

Entrada:

Problema generado con el generador aleatorio (2 habitaciones, 6 reservas).
(Problema en [A.5.1] de los Anexos)

Solución Esperada:

No creemos que se satisfazcan muchas reservas debido a que al ser generada al azar muchos días se solapan.

Solución Obtenida:

Como hemos explicado anteriormente no se han satisfecho todas las reservas debido a que algunas se solapan.
(Solución en [A.5.2] de los Anexos)

3.5.2 Prueba 2

Entrada:

Problema generado con el generador aleatorio (2 habitaciones, 6 reservas). Pero haciendo que ninguna fecha coincida. Además de que para cada habitación, mínimo una reserva coincida con el número de plazas de la habitación.
(Problema en [A.5.1] de los Anexos)

Solución Esperada:

Que todas las reservas se acumulen en una sola habitación. De este modo probamos que se priorice el hecho de dejar habitaciones libres antes de que haya plazas malgastadas.

Solución Obtenida:

Como hemos explicado anteriormente no se han satisfecho todas las reservas debido a que algunas se solapan.
(Solución en [A.5.2] de los Anexos)

4.Desarrollo de los modelos

Para el desarrollo del problema hemos hecho un diseño incremental, empezando por el nivel básico y a partir de ahí hemos ido ampliando el modelo con lo que se nos pedía en cada extensión. En el caso de la extensión 2 y 3 hemos usado de base la extensión uno y la hemos ampliado y en el caso de la extensión 4 hemos usado la tres y la hemos ampliado para que haga lo que se nos pide.

A parte de hacer el dominio para cada extensión, hemos hecho un generador para cada una que te genera un problema al azar con el número de habitaciones y de reservas que se le pase por parámetro.

5.Experimentación

Para la parte de experimentación hemos decidido ejecutar la extensión 1 incrementando cada vez el tamaño del problema, así podemos ver como evoluciona el tiempo de ejecución a medida que aumentamos el número de habitaciones y reservas.

Para seguir un método de aumento, hemos establecido la relación 1:5 entre habitaciones y reservas (i.e. cinco reservas por habitación). Por lo tanto cada ejecución incrementamos en uno el número de habitaciones y en cinco las reservas.

| Iteración | # habitaciones | # reservas | tiempo ejecución(s) | # estados evaluados |
|-----------|----------------|------------|---------------------|---------------------|
| 1 | 1 | 5 | 0,00 | 16 |
| 2 | 2 | 10 | 0,00 | 100 |
| 3 | 3 | 15 | 0,00 | 303 |
| 4 | 4 | 20 | 0'01 | 773 |
| 5 | 5 | 25 | 0,34 | 7028 |
| 6 | 6 | 30 | 10,00 | 29022 |
| 7 | 7 | 35 | 16,76 | 51763 |

Tiempo de Ejecución por Iteración

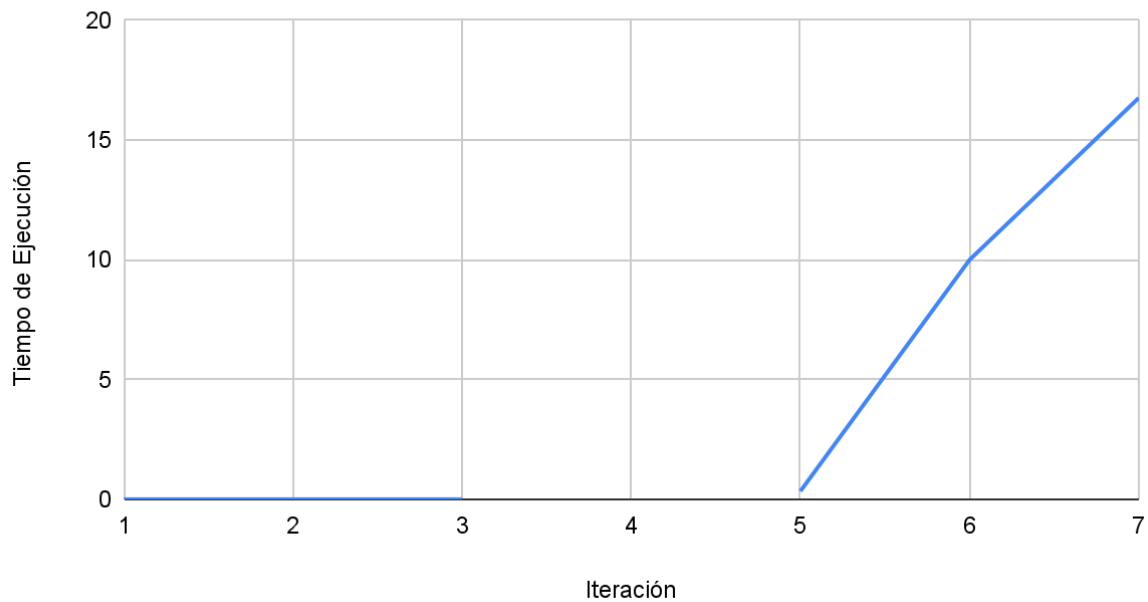


Figura 1: Gráfico de evolución del tiempo de ejecución

Estados Evaluados por Iteración

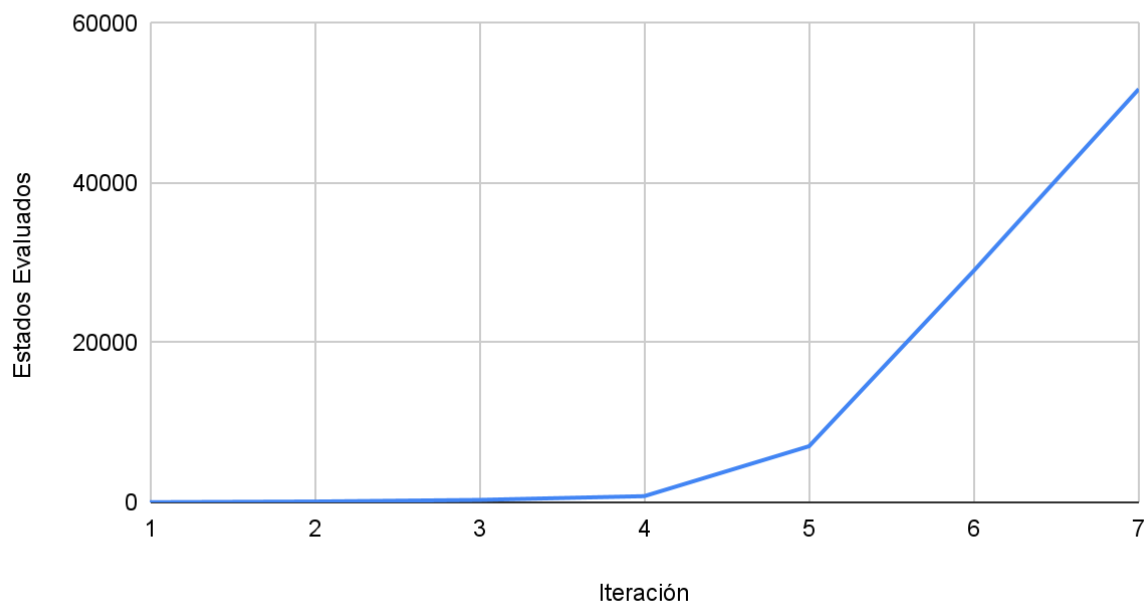


Figura 2: Gráfico de evolución de los estados evaluados

Como podemos ver en los gráficos, tanto el tiempo de ejecución como los estados evaluados crecen polinómicamente respecto al tamaño de la entrada, en la figura1, no se puede observar muy bien debido a que los tiempos de ejecución son muy bajos, en cambio en la figura2, se observa mejor la pendiente de crecimiento.

Hemos observado que a medida que crece el problema hay casos en los que tarda mucho porque explora muchos estados, pero también hay veces que explora muchos nodos y encuentra el óptimo en menos tiempo o veces que explora más nodos y tarda más en encontrar el óptimo. Por lo que podemos concluir que el tiempo de ejecución depende de la cantidad de nodos explorados y este depende del tamaño del problema.

(En el anexo [B. Anexo de Experimentación] se encuentran los problemas y las soluciones usadas para este apartado).

6. Conclusiones

Este trabajo nos ha servido para ver las funcionalidades que tienen los planificadores y para saber y entender cómo funcionan. Durante la elaboración del proyecto hemos ido reforzando los conocimientos obtenidos en clase, además de ser una ayuda para estudiar para los exámenes. Estamos satisfechos con el trabajo realizado y con los conocimientos obtenidos sobre las herramientas y lenguajes utilizados para el desarrollo de la IA.

A. Anexo de Juegos De Prueba

A.1 Juego de Pruebas Nivel Básico

A.1.1 Juego de prueba 1

A.1.1.1 Problema

(define (problem problemaBasico)

(:domain dominioBasico)

(:objects

hab1 hab2 -habitacion

res1 res2 res3 res4 res5 res6 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(= (plazas hab1) 4)

(= (plazas hab2) 4)

(= (personas res1) 1)

(= (inicio res1) 17)

(= (fin res1) 30)

(= (personas res2) 4)

(= (inicio res2) 11)

(= (fin res2) 19)

(= (personas res3) 4)

(= (inicio res3) 29)

(= (fin res3) 30)

(= (personas res4) 2)

(= (inicio res4) 5)

(= (fin res4) 10)

(= (personas res5) 4)

(= (inicio res5) 25)

(= (fin res5) 27)

(= (personas res6) 3)

(= (inicio res6) 6)

(= (fin res6) 20)

)

```
(:goal
  (forall (?res - reserva) (resuelta?res))
)
```

A.1.1.2 Solución

best first search space empty! problem proven unsolvable.

time spent: 0.00 seconds instantiating 0 easy, 12 hard action templates
 0.00 seconds reachability analysis, yielding 36 facts and 12 actions
 0.00 seconds creating final representation with 36 relevant facts, 0 relevant fluents
 0.00 seconds computing LNF
 0.00 seconds building connectivity graph
 0.00 seconds searching, evaluating 281 states, to a max depth of 1
 0.00 seconds total time

A.1.2 Juego de prueba 2

A.1.2.1 Problema

```
(define (problem problemaBasico)
  (:domain dominioBasico)

  (:objects
    hab1 hab2 -habitacion
    res1 res2 res3 res4 res5 res6 - reserva
  )

  (:init
    (not (resuelta res1))
    (not (resuelta res2))
    (not (resuelta res3))
    (not (resuelta res4))
    (not (resuelta res5))
    (not (resuelta res6))
    (= (plazas hab1) 4)
    (= (plazas hab2) 4)
```

```
(= (personas res1) 1)
(= (inicio res1) 10)
(= (fin res1) 13)
(= (personas res2) 4)
(= (inicio res2) 14)
(= (fin res2) 19)
(= (personas res3) 4)
(= (inicio res3) 20)
(= (fin res3) 24)
(= (personas res4) 2)
(= (inicio res4) 25)
(= (fin res4) 26)
(= (personas res5) 4)
(= (inicio res5) 27)
(= (fin res5) 28)
(= (personas res6) 3)
(= (inicio res6) 29)
(= (fin res6) 30)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
)
```

A1.2.2 Solución

```
step 0: RESERVAR_HABITACION RES6 HAB2
      1: RESERVAR_HABITACION RES5 HAB2
      2: RESERVAR_HABITACION RES4 HAB2
      3: RESERVAR_HABITACION RES3 HAB2
      4: RESERVAR_HABITACION RES2 HAB2
      5: RESERVAR_HABITACION RES1 HAB2
```

```
time spent: 0.00 seconds instantiating 0 easy, 12 hard action templates
            0.00 seconds reachability analysis, yielding 36 facts and 12 actions
            0.00 seconds creating final representation with 36 relevant facts, 0 relevant fluents
            0.00 seconds computing LNF
```

0.00 seconds building connectivity graph

0.00 seconds searching, evaluating 7 states, to a max depth of 1

0.00 seconds total time

A.2 Juego de Pruebas Extensión 1

A.2.1 Juego de prueba 1

A.2.1.1 Problema

(define (problem problemaExtension1)

(:domain dominioExtension1)

(:objects

hab1 hab2 -habitacion

res1 res2 res3 res4 res5 res6 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(= (reservasNoAsignadas) 0)

(= (plazas hab1) 1)

(= (plazas hab2) 4)

(= (personas res1) 1)

(= (inicio res1) 2)

(= (fin res1) 30)

(= (personas res2) 4)

(= (inicio res2) 22)

(= (fin res2) 25)

(= (personas res3) 3)

(= (inicio res3) 21)

(= (fin res3) 23)

(= (personas res4) 3)

(= (inicio res4) 6)

(= (fin res4) 15)

(= (personas res5) 3)

(= (inicio res5) 7)

(= (fin res5) 8)

(= (personas res6) 4)

(= (inicio res6) 24)

(= (fin res6) 30)

)

(:goal

(forall (?res - reserva) (resuelta?res))

```
)
(:metric minimize (reservasNoAsignadas))
)
```

A.2.1.2 Solución

step 0: RESERVAR_HABITACION RES6 HAB2

1: DENEGAR_RESERVA RES5 HAB1

2: DENEGAR_RESERVA RES4 HAB1

3: DENEGAR_RESERVA RES3 HAB1

4: DENEGAR_RESERVA RES2 HAB1

5: RESERVAR_HABITACION RES1 HAB1

time spent: 0.00 seconds instantiating 0 easy, 30 hard action templates

0.00 seconds reachability analysis, yielding 31 facts and 26 actions

0.00 seconds creating final representation with 26 relevant facts, 1 relevant fluents

0.00 seconds computing LNF

0.00 seconds building connectivity graph

0.00 seconds searching, evaluating 34 states, to a max depth of 0

0.00 seconds total time

A.2.2 Juego de prueba 2

A.2.2.1 Problema

(define (problem problemaExtension1)

(:domain dominioExtension1)

(:objects

hab1 -habitacion

res1 res2 res3 res4 res5 res6 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

```

(not (resuelta res6))
(= (reservasNoAsignadas) 0)
(= (plazas hab1) 4)
(= (personas res1) 1)
(= (inicio res1) 1)
(= (fin res1) 30)
(= (personas res2) 4)
(= (inicio res2) 2)
(= (fin res2) 3)
(= (personas res3) 3)
(= (inicio res3) 4)
(= (fin res3) 5)
(= (personas res4) 3)
(= (inicio res4) 6)
(= (fin res4) 7)
(= (personas res5) 3)
(= (inicio res5) 8)
(= (fin res5) 9)
(= (personas res6) 4)
(= (inicio res6) 24)
(= (fin res6) 25)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)

```

A.2.2.2 Solución

```

step 0: RESERVAR_HABITACION RES6 HAB1
      1: DENEGAR_RESERVA RES1 HAB1
      2: RESERVAR_HABITACION RES5 HAB1
      3: RESERVAR_HABITACION RES4 HAB1
      4: RESERVAR_HABITACION RES3 HAB1
      5: RESERVAR_HABITACION RES2 HAB1

```

time spent: 0.00 seconds instantiating 0 easy, 16 hard action templates

0.00 seconds reachability analysis, yielding 24 facts and 16 actions

0.00 seconds creating final representation with 24 relevant facts, 1 relevant fluents

0.00 seconds computing LNF

0.00 seconds building connectivity graph

0.01 seconds searching, evaluating 22 states, to a max depth of 0

0.01 seconds total time

A.3 Juego de Pruebas Extensión 2

A.3.1 Juego de prueba 1

A.3.1.1 Problema

(define (problem problemaExtension2)

(:domain dominioExtension2)

(:objects

hab1 hab2 -habitacion

res1 res2 res3 res4 res5 res6 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(= (total-cost) 0)

(= (num-res) 6)

(= (plazas hab1) 2)

(= (pcardinalhab hab1) 2)

(= (plazas hab2) 4)

(= (pcardinalhab hab2) 3)

(= (personas res1) 4)

(= (inicio res1) 3)

(= (fin res1) 4)

(= (pcardinalres res1) 3)

(= (personas res2) 2)

(= (inicio res2) 11)

(= (fin res2) 27)

(= (pcardinalres res2) 2)

(= (personas res3) 4)

(= (inicio res3) 12)

(= (fin res3) 18)

(= (pcardinalres res3) 1)

(= (personas res4) 2)

(= (inicio res4) 23)

(= (fin res4) 26)

(= (pcardinalres res4) 1)

(= (personas res5) 1)

(= (inicio res5) 19)

```
(= (fin res5) 28)
(= (pcardinalres res5) 2)
(= (personas res6) 2)
(= (inicio res6) 24)
(= (fin res6) 27)
(= (pcardinalres res6) 3)
)
(:goal
  (forall (?res - reserva) (resuelta ?res))
)
(:metric minimize (total-cost))
)
```

A.3.1.2 Solución

step 0: RESERVAR_HABITACION_PERFECTA RES1 HAB2

1: RESERVAR_HABITACION RES3 HAB2

2: RESERVAR_HABITACION_PERFECTA RES2 HAB1

3: DENEGAR_RESERVA RES4 HAB1

4: DENEGAR_RESERVA RES5 HAB1

5: RESERVAR_HABITACION_PERFECTA RES6 HAB2

time spent: 0.00 seconds instantiating 0 easy, 39 hard action templates

0.00 seconds reachability analysis, yielding 34 facts and 38 actions

0.00 seconds creating final representation with 32 relevant facts, 1 relevant fluents

0.00 seconds computing LNF

0.00 seconds building connectivity graph

0.00 seconds searching, evaluating 158 states, to a max depth of 0

0.00 seconds total time

A.3.2 Juego de prueba 2

A.3.2.1 Problema

(define (problem problemaExtension2)

(:domain dominioExtension2)

(:objects

hab1 hab2 -habitacion

res1 res2 res3 res4 res5 res6 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(= (total-cost) 0)

(= (num-res) 6)

(= (plazas hab1) 2)

(= (pcardinalhab hab1) 0)

(= (plazas hab2) 4)

(= (pcardinalhab hab2) 0)

(= (personas res1) 4)

(= (inicio res1) 3)

(= (fin res1) 4)

(= (pcardinalres res1) 3)

(= (personas res2) 2)

(= (inicio res2) 11)

(= (fin res2) 27)

(= (pcardinalres res2) 2)

(= (personas res3) 4)

(= (inicio res3) 12)

(= (fin res3) 18)

(= (pcardinalres res3) 1)

(= (personas res4) 2)

(= (inicio res4) 23)

(= (fin res4) 26)

(= (pcardinalres res4) 1)

(= (personas res5) 1)

(= (inicio res5) 19)

(= (fin res5) 28)

(= (pcardinalres res5) 2)

(= (personas res6) 2)


```
(= (inicio res6) 24)
(= (fin res6) 27)
(= (pcardinalres res6) 3)
)
(:goal
  (forall (?res - reserva) (resuelta ?res))
)
(:metric minimize (total-cost))
)
```

A.3.2.2 Solución

```
step 0: RESERVAR_HABITACION RES1 HAB2
      1: RESERVAR_HABITACION RES3 HAB2
      2: RESERVAR_HABITACION RES2 HAB1
      3: DENEGAR_RESERVA RES4 HAB1
      4: DENEGAR_RESERVA RES5 HAB1
      5: RESERVAR_HABITACION RES6 HAB2
```

```
time spent: 0.00 seconds instantiating 0 easy, 35 hard action templates
            0.00 seconds reachability analysis, yielding 34 facts and 34 actions
            0.00 seconds creating final representation with 32 relevant facts, 1 relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 94 states, to a max depth of 0
            0.00 seconds total time
```

A.4 Juego de Pruebas Extensión 3

A.4.1 Juego de prueba 1

A.4.1.1 Problema

(define (problem problemaExtension3)

(:domain dominioExtension3)

(:objects

hab1 hab2 - habitacion

res1 res2 res3 res4 res5 res6 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(= (reservasNoAsignadas) 0)

(= (plazasMalgastadas) 0)

(= (plazas hab1) 3)

(= (plazas hab2) 4)

(= (personas res1) 1)

(= (inicio res1) 20)

(= (fin res1) 26)

(= (personas res2) 2)

(= (inicio res2) 12)

(= (fin res2) 28)

(= (personas res3) 2)

(= (inicio res3) 21)

(= (fin res3) 24)

(= (personas res4) 4)

(= (inicio res4) 14)

(= (fin res4) 29)

(= (personas res5) 1)

(= (inicio res5) 11)

(= (fin res5) 28)

(= (personas res6) 2)

(= (inicio res6) 29)

(= (fin res6) 30)

)

(:goal

```
(forall (?res - reserva) (resuelta?res))
)
(:metric minimize (+ (* (reservasNoAsignadas) 18) (plazasMalgastadas)))
)
```

A.4.1.2 Solución

```
step 0: DENEGAR_RESERVA RES4 HAB1
      1: RESERVAR_HABITACION RES6 HAB1
      2: RESERVAR_HABITACION RES5 HAB2
      3: DENEGAR_RESERVA RES1 HAB2
      4: DENEGAR_RESERVA RES3 HAB2
      5: RESERVAR_HABITACION RES2 HAB1
```

```
time spent: 0.00 seconds instantiating 0 easy, 44 hard action templates
            0.00 seconds reachability analysis, yielding 35 facts and 40 actions
            0.00 seconds creating final representation with 34 relevant facts, 2 relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 65 states, to a max depth of 0
            0.00 seconds total time
```

A.4.2 Juego de prueba 2

A.4.2.1 Problema

```
(define (problem problemaExtension3)
  (:domain dominioExtension3)

  (:objects
    hab1 hab2 - habitacion
    res1 res2 res3 res4 res5 res6 - reserva
  )

  (:init
    (not (resuelta res1))
    (not (resuelta res2))
    (not (resuelta res3))
    (not (resuelta res4))
    (not (resuelta res5))
    (not (resuelta res6))
    (= (reservasNoAsignadas) 0)
    (= (plazasMalgastadas) 0)
    (= (plazas hab1) 4)
    (= (plazas hab2) 4)
    (= (personas res1) 1)
    (= (inicio res1) 2)
```

```

(= (fin res1) 5)
(= (personas res2) 2)
(= (inicio res2) 2)
(= (fin res2) 5)
(= (personas res3) 2)
(= (inicio res3) 10)
(= (fin res3) 14)
(= (personas res4) 4)
(= (inicio res4) 15)
(= (fin res4) 17)
(= (personas res5) 1)
(= (inicio res5) 18)
(= (fin res5) 19)
(= (personas res6) 2)
(= (inicio res6) 29)
(= (fin res6) 30)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (+ (* (reservasNoAsignadas) 18) (plazasMalgastadas)))
)

```

A.4.2.2 Solución

```

step 0: RESERVAR_HABITACION RES5 HAB1
1: RESERVAR_HABITACION RES1 HAB1
2: RESERVAR_HABITACION RES6 HAB1
3: RESERVAR_HABITACION RES3 HAB1
4: RESERVAR_HABITACION RES2 HAB2
5: RESERVAR_HABITACION RES4 HAB1

```

```

time spent: 0.00 seconds instantiating 0 easy, 12 hard action templates
0.00 seconds reachability analysis, yielding 36 facts and 12 actions
0.00 seconds creating final representation with 36 relevant facts, 2 relevant fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.00 seconds searching, evaluating 40 states, to a max depth of 0
0.00 seconds total time

```


A.5 Juego de Pruebas Extensión 4

A.5.1 Juego de prueba 1

A.5.1.1 Problema

```
(define (problem problemaExtension4)
  (:domain dominioExtension4)
  (:objects
    hab1 hab2 - habitacion
    res1 res2 res3 res4 res5 res6 - reserva
  )
  (:init
    (not (resuelta res1))
    (not (resuelta res2))
    (not (resuelta res3))
    (not (resuelta res4))
    (not (resuelta res5))
    (not (resuelta res6))
    (= (reservasNoAsignadas) 0)
    (= (plazasMalgastadas) 0)
    (= (habitacionesDiferentes) 0)
    (= (plazas hab1) 4)
    (= (plazas hab2) 1)
    (= (personas res1) 1)
    (= (inicio res1) 1)
    (= (fin res1) 2)
    (= (personas res2) 4)
    (= (inicio res2) 3)
    (= (fin res2) 4)
    (= (personas res3) 3)
    (= (inicio res3) 5)
    (= (fin res3) 6)
    (= (personas res4) 3)
    (= (inicio res4) 7)
    (= (fin res4) 8)
    (= (personas res5) 2)
    (= (inicio res5) 9)
    (= (fin res5) 10)
    (= (personas res6) 1)
    (= (inicio res6) 17)
    (= (fin res6) 28)
  )
  (:goal
    (forall (?res - reserva) (resuelta?res))
  )
  (:metric minimize (+ (+ (* (reservasNoAsignadas) 54) (plazasMalgastadas)) (*
    (habitacionesDiferentes) 18)))
)
```

A.5.1.2 Solución

```

step  0: DENEGAR_RESERVA RES2 HAB2
      1: RESERVAR_HABITACION_NO_USADA RES1 HAB1
      2: RESERVAR_HABITACION_NO_USADA RES5 HAB2
      3: DENEGAR_RESERVA RES3 HAB2
      4: DENEGAR_RESERVA RES4 HAB2
      5: RESERVAR_HABITACION RES6 HAB1

```

```

time spent:  0.00 seconds instantiating 0 easy, 48 hard action templates
            0.00 seconds reachability analysis, yielding 38 facts and 46 actions
            0.00 seconds creating final representation with 36 relevant facts, 3 relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 63 states, to a max depth of 0
            0.00 seconds total time

```

A.5.2 Juego de prueba 2

A.5.2.1 Problema

```

(define (problem problemaExtension4)
  (:domain dominioExtension4)
  (:objects
    hab1 hab2 - habitacion
    res1 res2 res3 res4 res5 res6 - reserva
  )
  (:init
    (not (resuelta res1))
    (not (resuelta res2))
    (not (resuelta res3))
    (not (resuelta res4))
    (not (resuelta res5))
    (not (resuelta res6))
    (= (reservasNoAsignadas) 0)
    (= (plazasMalgastadas) 0)
    (= (habitacionesDiferentes) 0)
    (= (plazas hab1) 4)
    (= (plazas hab2) 1)
    (= (personas res1) 1)
    (= (inicio res1) 1)
    (= (fin res1) 2)
    (= (personas res2) 4)
    (= (inicio res2) 3)
    (= (fin res2) 4)
    (= (personas res3) 3)
    (= (inicio res3) 5)
    (= (fin res3) 6)
    (= (personas res4) 3)
  )

```

```
(= (inicio res4) 7)
(= (fin res4) 8)
(= (personas res5) 2)
(= (inicio res5) 9)
(= (fin res5) 10)
(= (personas res6) 1)
(= (inicio res6) 11)
(= (fin res6) 12)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
  (:metric minimize (+ (+ (* (reservasNoAsignadas) 54) (plazasMalgastadas)) (*
(habitacionesDiferentes) 18)))
)
```

A.5.2.2 Solución

```
step 0: RESERVAR_HABITACION_NO_USADA RES6 HAB1
1: RESERVAR_HABITACION RES1 HAB1
2: RESERVAR_HABITACION RES5 HAB1
3: RESERVAR_HABITACION RES3 HAB1
4: RESERVAR_HABITACION RES4 HAB1
5: RESERVAR_HABITACION RES2 HAB1
```

```
time spent: 0.00 seconds instantiating 0 easy, 20 hard action templates
           0.00 seconds reachability analysis, yielding 36 facts and 20 actions
           0.00 seconds creating final representation with 32 relevant facts, 3 relevant fluents
           0.00 seconds computing LNF
           0.00 seconds building connectivity graph
           0.00 seconds searching, evaluating 43 states, to a max depth of 0
           0.00 seconds total time
```


B. Anexo de Experimentación

B.1 (1 habitación 5 reservas)

B.1.1 Problema

```
(define (problem problemaExtension1)
  (:domain dominioExtension1)
```

```
  (:objects
    hab1 -habitacion
    res1 res2 res3 res4 res5 - reserva
  )
```

```
  (:init
    (not (resuelta res1))
    (not (resuelta res2))
    (not (resuelta res3))
    (not (resuelta res4))
    (not (resuelta res5))
    (= (reservasNoAsignadas) 0)
    (= (plazas hab1) 1)
    (= (personas res1) 3)
    (= (inicio res1) 18)
    (= (fin res1) 20)
    (= (personas res2) 2)
    (= (inicio res2) 2)
    (= (fin res2) 8)
    (= (personas res3) 4)
    (= (inicio res3) 11)
    (= (fin res3) 26)
    (= (personas res4) 2)
    (= (inicio res4) 22)
    (= (fin res4) 28)
    (= (personas res5) 3)
    (= (inicio res5) 25)
    (= (fin res5) 29)
  )
  (:goal
    (forall (?res - reserva) (resuelta?res))
  )
  (:metric minimize (reservasNoAsignadas))
)
```

B.1.2 Solución

step 0: DENEGAR_RESERVA RES5 HAB1

1: DENEGAR_RESERVA RES4 HAB1

2: DENEGAR_RESERVA RES3 HAB1

3: DENEGAR_RESERVA RES2 HAB1

4: DENEGAR_RESERVA RES1 HAB1

time spent: 0.00 seconds instantiating 0 easy, 5 hard action templates

0.00 seconds reachability analysis, yielding 15 facts and 5 actions

0.00 seconds creating final representation with 10 relevant facts, 1 relevant fluents

0.00 seconds computing LNF

0.00 seconds building connectivity graph

0.00 seconds searching, evaluating 16 states, to a max depth of 0

0.00 seconds total time

B.2 (2 habitaciones 10 reserva)

B.2.1 Problema

(define (problem problemaExtension1)

(:domain dominioExtension1)

(:objects

hab1 hab2 -habitacion

res1 res2 res3 res4 res5 res6 res7 res8 res9 res10 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(not (resuelta res7))

(not (resuelta res8))

(not (resuelta res9))

(not (resuelta res10))

(= (reservasNoAsignadas) 0)

(= (plazas hab1) 1)

(= (plazas hab2) 3)

(= (personas res1) 1)

(= (inicio res1) 14)

(= (fin res1) 29)

(= (personas res2) 4)

(= (inicio res2) 14)

(= (fin res2) 21)

(= (personas res3) 4)

(= (inicio res3) 25)

(= (fin res3) 29)

(= (personas res4) 4)

(= (inicio res4) 19)

(= (fin res4) 24)

(= (personas res5) 2)

(= (inicio res5) 20)

(= (fin res5) 26)

(= (personas res6) 1)

(= (inicio res6) 4)

(= (fin res6) 5)

(= (personas res7) 2)

(= (inicio res7) 18)

```
(= (fin res7) 30)
(= (personas res8) 2)
(= (inicio res8) 27)
(= (fin res8) 29)
(= (personas res9) 4)
(= (inicio res9) 7)
(= (fin res9) 21)
(= (personas res10) 1)
(= (inicio res10) 21)
(= (fin res10) 26)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)
```

B.2.2 Solución

ff: found legal plan as follows

```
step 0: RESERVAR_HABITACION RES8 HAB2
1: DENEGAR_RESERVA RES9 HAB2
2: DENEGAR_RESERVA RES7 HAB1
3: DENEGAR_RESERVA RES5 HAB1
4: DENEGAR_RESERVA RES4 HAB2
5: DENEGAR_RESERVA RES3 HAB2
6: DENEGAR_RESERVA RES2 HAB2
7: RESERVAR_HABITACION RES6 HAB2
8: RESERVAR_HABITACION RES10 HAB1
9: DENEGAR_RESERVA RES1 HAB1
```

```
time spent: 0.00 seconds instantiating 0 easy, 55 hard action templates
            0.00 seconds reachability analysis, yielding 49 facts and 34 actions
            0.00 seconds creating final representation with 38 relevant facts, 1 relevant fluents
            0.00 seconds computing LNF
            0.00 seconds building connectivity graph
            0.00 seconds searching, evaluating 100 states, to a max depth of 0
            0.00 seconds total time
```

B.3 (3 habitaciones 15 reserva)

B.3.1 Problema

(define (problem problemaExtension1)

(:domain dominioExtension1)

(:objects

hab1 hab2 hab3 -habitacion

res1 res2 res3 res4 res5 res6 res7 res8 res9 res10 res11 res12 res13 res14 res15 -
reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(not (resuelta res7))

(not (resuelta res8))

(not (resuelta res9))

(not (resuelta res10))

(not (resuelta res11))

(not (resuelta res12))

(not (resuelta res13))

(not (resuelta res14))

(not (resuelta res15))

(= (reservasNoAsignadas) 0)

(= (plazas hab1) 3)

(= (plazas hab2) 1)

(= (plazas hab3) 3)

(= (personas res1) 3)

(= (inicio res1) 22)

(= (fin res1) 27)

(= (personas res2) 4)

(= (inicio res2) 12)

(= (fin res2) 24)

(= (personas res3) 3)

(= (inicio res3) 15)

(= (fin res3) 21)

(= (personas res4) 4)

(= (inicio res4) 21)

(= (fin res4) 29)

(= (personas res5) 4)

(= (inicio res5) 16)

(= (fin res5) 17)

(= (personas res6) 3)

(= (inicio res6) 2)

(= (fin res6) 20)

(= (personas res7) 2)

(= (inicio res7) 15)

```

(= (fin res7) 22)
(= (personas res8) 4)
(= (inicio res8) 26)
(= (fin res8) 30)
(= (personas res9) 4)
(= (inicio res9) 13)
(= (fin res9) 27)
(= (personas res10) 4)
(= (inicio res10) 22)
(= (fin res10) 28)
(= (personas res11) 1)
(= (inicio res11) 12)
(= (fin res11) 27)
(= (personas res12) 2)
(= (inicio res12) 20)
(= (fin res12) 22)
(= (personas res13) 1)
(= (inicio res13) 3)
(= (fin res13) 5)
(= (personas res14) 3)
(= (inicio res14) 21)
(= (fin res14) 30)
(= (personas res15) 1)
(= (inicio res15) 9)
(= (fin res15) 18)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)

```

B.3.2 Solución

```

step 0: RESERVAR_HABITACION RES1 HAB1
1: DENEGAR_RESERVA RES14 HAB2
2: DENEGAR_RESERVA RES12 HAB2
3: DENEGAR_RESERVA RES10 HAB3
4: DENEGAR_RESERVA RES9 HAB3
5: DENEGAR_RESERVA RES8 HAB3
6: DENEGAR_RESERVA RES7 HAB2
7: DENEGAR_RESERVA RES6 HAB2
8: DENEGAR_RESERVA RES5 HAB3
9: DENEGAR_RESERVA RES4 HAB3
10: DENEGAR_RESERVA RES3 HAB2
11: DENEGAR_RESERVA RES2 HAB3
12: RESERVAR_HABITACION RES13 HAB3
13: RESERVAR_HABITACION RES15 HAB1
14: RESERVAR_HABITACION RES11 HAB2

```

time spent: 0.00 seconds instantiating 0 easy, 190 hard action templates

0.00 seconds reachability analysis, yielding 96 facts and 115 actions
 0.00 seconds creating final representation with 72 relevant facts, 1 relevant fluents
 0.00 seconds computing LNF
 0.00 seconds building connectivity graph
 0.01 seconds searching, evaluating 303 states, to a max depth of 0
 0.01 seconds total time

B.4 (4 habitaciones 20 reserva)

B.4.1 Problema

```
(define (problem problemaExtension1)
```

```
  (:domain dominioExtension1)
```

```
  (:objects
```

```
    hab1 hab2 hab3 hab4 -habitacion
```

```
    res1 res2 res3 res4 res5 res6 res7 res8 res9 res10 res11 res12 res13 res14 res15 res16  
    res17 res18 res19 res20 - reserva  
  )
```

```
  (:init
```

```
    (not (resuelta res1))
```

```
    (not (resuelta res2))
```

```
    (not (resuelta res3))
```

```
    (not (resuelta res4))
```

```
    (not (resuelta res5))
```

```
    (not (resuelta res6))
```

```
    (not (resuelta res7))
```

```
    (not (resuelta res8))
```

```
    (not (resuelta res9))
```

```
    (not (resuelta res10))
```

```
    (not (resuelta res11))
```

```
    (not (resuelta res12))
```

```
    (not (resuelta res13))
```

```
    (not (resuelta res14))
```

```
    (not (resuelta res15))
```

```
    (not (resuelta res16))
```

```
    (not (resuelta res17))
```

```
    (not (resuelta res18))
```

```
    (not (resuelta res19))
```

```
    (not (resuelta res20))
```

```
    (= (reservasNoAsignadas) 0)
```

```
    (= (plazas hab1) 2)
```

```
    (= (plazas hab2) 2)
```

```
    (= (plazas hab3) 4)
```

```
    (= (plazas hab4) 1)
```

```
    (= (personas res1) 4)
```

```
    (= (inicio res1) 7)
```

```
    (= (fin res1) 23)
```

```
    (= (personas res2) 3)
```

```
    (= (inicio res2) 15)
```

(= (fin res2) 18)
 (= (personas res3) 4)
 (= (inicio res3) 16)
 (= (fin res3) 20)
 (= (personas res4) 1)
 (= (inicio res4) 22)
 (= (fin res4) 27)
 (= (personas res5) 2)
 (= (inicio res5) 7)
 (= (fin res5) 27)
 (= (personas res6) 4)
 (= (inicio res6) 17)
 (= (fin res6) 19)
 (= (personas res7) 4)
 (= (inicio res7) 9)
 (= (fin res7) 25)
 (= (personas res8) 1)
 (= (inicio res8) 15)
 (= (fin res8) 24)
 (= (personas res9) 2)
 (= (inicio res9) 15)
 (= (fin res9) 27)
 (= (personas res10) 1)
 (= (inicio res10) 12)
 (= (fin res10) 24)
 (= (personas res11) 2)
 (= (inicio res11) 23)
 (= (fin res11) 25)
 (= (personas res12) 3)
 (= (inicio res12) 20)
 (= (fin res12) 23)
 (= (personas res13) 2)
 (= (inicio res13) 6)
 (= (fin res13) 16)
 (= (personas res14) 3)
 (= (inicio res14) 14)
 (= (fin res14) 27)
 (= (personas res15) 3)
 (= (inicio res15) 8)
 (= (fin res15) 26)
 (= (personas res16) 2)
 (= (inicio res16) 23)
 (= (fin res16) 27)
 (= (personas res17) 1)
 (= (inicio res17) 26)
 (= (fin res17) 27)
 (= (personas res18) 4)
 (= (inicio res18) 16)
 (= (fin res18) 29)
 (= (personas res19) 1)
 (= (inicio res19) 16)
 (= (fin res19) 28)
 (= (personas res20) 2)


```
(= (inicio res20) 9)
(= (fin res20) 25)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)
```

B.4.2 Solución

```
step 0: DENEGAR_RESERVA RES20 HAB4
1: DENEGAR_RESERVA RES18 HAB4
2: DENEGAR_RESERVA RES16 HAB4
3: DENEGAR_RESERVA RES15 HAB4
4: DENEGAR_RESERVA RES14 HAB4
5: DENEGAR_RESERVA RES13 HAB4
6: DENEGAR_RESERVA RES12 HAB4
7: DENEGAR_RESERVA RES11 HAB4
8: DENEGAR_RESERVA RES9 HAB4
9: DENEGAR_RESERVA RES7 HAB4
10: DENEGAR_RESERVA RES6 HAB4
11: DENEGAR_RESERVA RES5 HAB4
12: DENEGAR_RESERVA RES3 HAB4
13: DENEGAR_RESERVA RES2 HAB4
14: DENEGAR_RESERVA RES1 HAB4
15: RESERVAR_HABITACION RES17 HAB4
16: DENEGAR_RESERVA RES19 HAB4
17: RESERVAR_HABITACION RES10 HAB1
18: DENEGAR_RESERVA RES4 HAB1
19: RESERVAR_HABITACION RES8 HAB2
```

```
time spent: 0.00 seconds instantiating 0 easy, 684 hard action templates
            0.00 seconds reachability analysis, yielding 167 facts and 496 actions
            0.00 seconds creating final representation with 134 relevant facts, 1 relevant
```

```
fluents
0.00 seconds computing LNF
0.00 seconds building connectivity graph
0.01 seconds searching, evaluating 773 states, to a max depth of 0
0.01 seconds total time
```

B.5 (5 habitaciones 25 reserva)

B.5.1 Problema

(define (problem problemaExtension1)

(:domain dominioExtension1)

(:objects

hab1 hab2 hab3 hab4 hab5 -habitacion

res1 res2 res3 res4 res5 res6 res7 res8 res9 res10 res11 res12 res13 res14 res15 res16
res17 res18 res19 res20 res21 res22 res23 res24 res25 - reserva

)

(:init

(not (resuelta res1))

(not (resuelta res2))

(not (resuelta res3))

(not (resuelta res4))

(not (resuelta res5))

(not (resuelta res6))

(not (resuelta res7))

(not (resuelta res8))

(not (resuelta res9))

(not (resuelta res10))

(not (resuelta res11))

(not (resuelta res12))

(not (resuelta res13))

(not (resuelta res14))

(not (resuelta res15))

(not (resuelta res16))

(not (resuelta res17))

(not (resuelta res18))

(not (resuelta res19))

(not (resuelta res20))

(not (resuelta res21))

(not (resuelta res22))

(not (resuelta res23))

(not (resuelta res24))

(not (resuelta res25))

(= (reservasNoAsignadas) 0)

(= (plazas hab1) 1)

(= (plazas hab2) 3)

(= (plazas hab3) 2)

(= (plazas hab4) 4)

(= (plazas hab5) 4)

(= (personas res1) 2)

(= (inicio res1) 2)

(= (fin res1) 25)

(= (personas res2) 3)

(= (inicio res2) 15)

(= (fin res2) 20)

(= (personas res3) 4)

(= (inicio res3) 11)
 (= (fin res3) 19)
 (= (personas res4) 3)
 (= (inicio res4) 26)
 (= (fin res4) 29)
 (= (personas res5) 4)
 (= (inicio res5) 9)
 (= (fin res5) 16)
 (= (personas res6) 1)
 (= (inicio res6) 10)
 (= (fin res6) 28)
 (= (personas res7) 3)
 (= (inicio res7) 14)
 (= (fin res7) 24)
 (= (personas res8) 2)
 (= (inicio res8) 16)
 (= (fin res8) 23)
 (= (personas res9) 1)
 (= (inicio res9) 18)
 (= (fin res9) 23)
 (= (personas res10) 3)
 (= (inicio res10) 23)
 (= (fin res10) 26)
 (= (personas res11) 2)
 (= (inicio res11) 24)
 (= (fin res11) 27)
 (= (personas res12) 1)
 (= (inicio res12) 25)
 (= (fin res12) 28)
 (= (personas res13) 3)
 (= (inicio res13) 4)
 (= (fin res13) 25)
 (= (personas res14) 1)
 (= (inicio res14) 20)
 (= (fin res14) 21)
 (= (personas res15) 2)
 (= (inicio res15) 9)
 (= (fin res15) 27)
 (= (personas res16) 3)
 (= (inicio res16) 12)
 (= (fin res16) 21)
 (= (personas res17) 4)
 (= (inicio res17) 12)
 (= (fin res17) 17)
 (= (personas res18) 1)
 (= (inicio res18) 3)
 (= (fin res18) 22)
 (= (personas res19) 1)
 (= (inicio res19) 27)
 (= (fin res19) 30)
 (= (personas res20) 1)
 (= (inicio res20) 19)
 (= (fin res20) 20)

```
(= (personas res21) 1)
(= (inicio res21) 11)
(= (fin res21) 21)
(= (personas res22) 4)
(= (inicio res22) 15)
(= (fin res22) 22)
(= (personas res23) 2)
(= (inicio res23) 19)
(= (fin res23) 25)
(= (personas res24) 4)
(= (inicio res24) 21)
(= (fin res24) 28)
(= (personas res25) 2)
(= (inicio res25) 29)
(= (fin res25) 30)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)
```

B.5.2 Solución

```
step 0: RESERVAR_HABITACION RES25 HAB2
1: DENEGAR_RESERVA RES24 HAB3
2: DENEGAR_RESERVA RES23 HAB1
3: DENEGAR_RESERVA RES22 HAB3
4: DENEGAR_RESERVA RES17 HAB3
5: DENEGAR_RESERVA RES16 HAB3
6: DENEGAR_RESERVA RES15 HAB1
7: DENEGAR_RESERVA RES13 HAB3
8: DENEGAR_RESERVA RES11 HAB1
9: DENEGAR_RESERVA RES10 HAB3
10: DENEGAR_RESERVA RES8 HAB1
11: DENEGAR_RESERVA RES7 HAB3
12: DENEGAR_RESERVA RES5 HAB3
13: DENEGAR_RESERVA RES4 HAB3
14: DENEGAR_RESERVA RES3 HAB3
15: DENEGAR_RESERVA RES2 HAB3
16: DENEGAR_RESERVA RES1 HAB1
17: RESERVAR_HABITACION RES12 HAB5
18: RESERVAR_HABITACION RES19 HAB4
19: DENEGAR_RESERVA RES6 HAB4
20: RESERVAR_HABITACION RES14 HAB1
21: DENEGAR_RESERVA RES9 HAB1
22: DENEGAR_RESERVA RES18 HAB1
23: RESERVAR_HABITACION RES21 HAB2
24: RESERVAR_HABITACION RES20 HAB3
```

time spent: 0.01 seconds instantiating 0 easy, 1316 hard action templates

```

0.00 seconds reachability analysis, yielding 267 facts and 1111 actions
0.00 seconds creating final representation with 234 relevant facts, 1 relevant
fluents
0.01 seconds computing LNF
0.00 seconds building connectivity graph
0.32 seconds searching, evaluating 7028 states, to a max depth of 0
0.34 seconds total time

```

B.6 (6 habitaciones 30 reserva)

B.6.1 Problema

```

(define (problem problemaExtension1)
  (:domain dominioExtension1)

```

```

  (:objects
    hab1 hab2 hab3 hab4 hab5 hab6 -habitacion
    res1 res2 res3 res4 res5 res6 res7 res8 res9 res10 res11 res12 res13 res14 res15 res16
    res17 res18 res19 res20 res21 res22 res23 res24 res25 res26 res27 res28 res29 res30 -
    reserva
  )

```

```

  (:init
    (not (resuelta res1))
    (not (resuelta res2))
    (not (resuelta res3))
    (not (resuelta res4))
    (not (resuelta res5))
    (not (resuelta res6))
    (not (resuelta res7))
    (not (resuelta res8))
    (not (resuelta res9))
    (not (resuelta res10))
    (not (resuelta res11))
    (not (resuelta res12))
    (not (resuelta res13))
    (not (resuelta res14))
    (not (resuelta res15))
    (not (resuelta res16))
    (not (resuelta res17))
    (not (resuelta res18))
    (not (resuelta res19))
    (not (resuelta res20))
    (not (resuelta res21))
    (not (resuelta res22))
    (not (resuelta res23))
    (not (resuelta res24))
    (not (resuelta res25))
    (not (resuelta res26))
    (not (resuelta res27))
    (not (resuelta res28))
  )

```

(not (resuelta res29))
(not (resuelta res30))
(= (reservasNoAsignadas) 0)
(= (plazas hab1) 2)
(= (plazas hab2) 1)
(= (plazas hab3) 4)
(= (plazas hab4) 2)
(= (plazas hab5) 3)
(= (plazas hab6) 4)
(= (personas res1) 4)
(= (inicio res1) 15)
(= (fin res1) 28)
(= (personas res2) 4)
(= (inicio res2) 16)
(= (fin res2) 17)
(= (personas res3) 3)
(= (inicio res3) 29)
(= (fin res3) 30)
(= (personas res4) 4)
(= (inicio res4) 4)
(= (fin res4) 22)
(= (personas res5) 1)
(= (inicio res5) 20)
(= (fin res5) 22)
(= (personas res6) 3)
(= (inicio res6) 14)
(= (fin res6) 26)
(= (personas res7) 3)
(= (inicio res7) 8)
(= (fin res7) 21)
(= (personas res8) 1)
(= (inicio res8) 26)
(= (fin res8) 30)
(= (personas res9) 3)
(= (inicio res9) 12)
(= (fin res9) 13)
(= (personas res10) 1)
(= (inicio res10) 4)
(= (fin res10) 24)
(= (personas res11) 1)
(= (inicio res11) 28)
(= (fin res11) 29)
(= (personas res12) 3)
(= (inicio res12) 26)
(= (fin res12) 28)
(= (personas res13) 1)
(= (inicio res13) 23)
(= (fin res13) 26)
(= (personas res14) 4)
(= (inicio res14) 9)
(= (fin res14) 26)
(= (personas res15) 1)
(= (inicio res15) 27)

```

(= (fin res15) 29)
(= (personas res16) 4)
(= (inicio res16) 4)
(= (fin res16) 17)
(= (personas res17) 1)
(= (inicio res17) 26)
(= (fin res17) 27)
(= (personas res18) 4)
(= (inicio res18) 17)
(= (fin res18) 30)
(= (personas res19) 3)
(= (inicio res19) 12)
(= (fin res19) 26)
(= (personas res20) 2)
(= (inicio res20) 20)
(= (fin res20) 24)
(= (personas res21) 3)
(= (inicio res21) 14)
(= (fin res21) 21)
(= (personas res22) 4)
(= (inicio res22) 22)
(= (fin res22) 26)
(= (personas res23) 2)
(= (inicio res23) 14)
(= (fin res23) 27)
(= (personas res24) 2)
(= (inicio res24) 16)
(= (fin res24) 30)
(= (personas res25) 4)
(= (inicio res25) 14)
(= (fin res25) 28)
(= (personas res26) 2)
(= (inicio res26) 12)
(= (fin res26) 14)
(= (personas res27) 3)
(= (inicio res27) 20)
(= (fin res27) 23)
(= (personas res28) 4)
(= (inicio res28) 24)
(= (fin res28) 25)
(= (personas res29) 2)
(= (inicio res29) 14)
(= (fin res29) 20)
(= (personas res30) 1)
(= (inicio res30) 7)
(= (fin res30) 24)
)
(:goal
  (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)

```

B.6.2 Solución

step 0: RESERVAR_HABITACION RES3 HAB3

- 1: DENEGAR_RESERVA RES29 HAB2
- 2: DENEGAR_RESERVA RES28 HAB5
- 3: DENEGAR_RESERVA RES27 HAB4
- 4: DENEGAR_RESERVA RES26 HAB2
- 5: DENEGAR_RESERVA RES25 HAB5
- 6: DENEGAR_RESERVA RES24 HAB2
- 7: DENEGAR_RESERVA RES23 HAB2
- 8: DENEGAR_RESERVA RES22 HAB5
- 9: DENEGAR_RESERVA RES21 HAB4
- 10: DENEGAR_RESERVA RES20 HAB2
- 11: DENEGAR_RESERVA RES19 HAB4
- 12: DENEGAR_RESERVA RES18 HAB5
- 13: DENEGAR_RESERVA RES16 HAB5
- 14: DENEGAR_RESERVA RES14 HAB5
- 15: DENEGAR_RESERVA RES12 HAB4
- 16: DENEGAR_RESERVA RES9 HAB4
- 17: DENEGAR_RESERVA RES7 HAB4
- 18: DENEGAR_RESERVA RES6 HAB4
- 19: DENEGAR_RESERVA RES2 HAB5
- 20: DENEGAR_RESERVA RES1 HAB5
- 21: RESERVAR_HABITACION RES17 HAB1
- 22: RESERVAR_HABITACION RES15 HAB2
- 23: RESERVAR_HABITACION RES11 HAB1
- 24: RESERVAR_HABITACION RES8 HAB6
- 25: RESERVAR_HABITACION RES4 HAB3
- 26: DENEGAR_RESERVA RES10 HAB3
- 27: DENEGAR_RESERVA RES30 HAB3
- 28: RESERVAR_HABITACION RES13 HAB2
- 29: RESERVAR_HABITACION RES5 HAB1

time spent: 0.01 seconds instantiating 0 easy, 1729 hard action templates

0.00 seconds reachability analysis, yielding 355 facts and 1334 actions

0.01 seconds creating final representation with 290 relevant facts, 1 relevant

fluents

0.00 seconds computing LNF

0.00 seconds building connectivity graph

9.98 seconds searching, evaluating 29022 states, to a max depth of 0

10.00 seconds total time

B.7 (7 habitaciones 35 reserva)

B.7.1 Problema

```
(define (problem problemaExtension1)
```

```
(:domain dominioExtension1)
```

```
(:objects
```

```
  hab1 hab2 hab3 hab4 hab5 hab6 hab7 -habitacion
```

```
  res1 res2 res3 res4 res5 res6 res7 res8 res9 res10 res11 res12 res13 res14 res15 res16
```

```
res17 res18 res19 res20 res21 res22 res23 res24 res25 res26 res27 res28 res29 res30
```

```
res31 res32 res33 res34 res35 - reserva
```

```
)
```

```
(:init
```

```
  (not (resuelta res1))
```

```
  (not (resuelta res2))
```

```
  (not (resuelta res3))
```

```
  (not (resuelta res4))
```

```
  (not (resuelta res5))
```

```
  (not (resuelta res6))
```

```
  (not (resuelta res7))
```

```
  (not (resuelta res8))
```

```
  (not (resuelta res9))
```

```
  (not (resuelta res10))
```

```
  (not (resuelta res11))
```

```
  (not (resuelta res12))
```

```
  (not (resuelta res13))
```

```
  (not (resuelta res14))
```

```
  (not (resuelta res15))
```

```
  (not (resuelta res16))
```

```
  (not (resuelta res17))
```

```
  (not (resuelta res18))
```

```
  (not (resuelta res19))
```

```
  (not (resuelta res20))
```

```
  (not (resuelta res21))
```

```
  (not (resuelta res22))
```

```
  (not (resuelta res23))
```

```
  (not (resuelta res24))
```

```
  (not (resuelta res25))
```

```
  (not (resuelta res26))
```

```
  (not (resuelta res27))
```

```
  (not (resuelta res28))
```

```
  (not (resuelta res29))
```

```
  (not (resuelta res30))
```

```
  (not (resuelta res31))
```

```
  (not (resuelta res32))
```

```
  (not (resuelta res33))
```

```
  (not (resuelta res34))
```

```
  (not (resuelta res35))
```

```
  (= (reservasNoAsignadas) 0)
```

```
  (= (plazas hab1) 3)
```

(= (plazas hab2) 3)
 (= (plazas hab3) 1)
 (= (plazas hab4) 2)
 (= (plazas hab5) 2)
 (= (plazas hab6) 3)
 (= (plazas hab7) 1)
 (= (personas res1) 1)
 (= (inicio res1) 20)
 (= (fin res1) 27)
 (= (personas res2) 3)
 (= (inicio res2) 17)
 (= (fin res2) 20)
 (= (personas res3) 1)
 (= (inicio res3) 26)
 (= (fin res3) 29)
 (= (personas res4) 2)
 (= (inicio res4) 27)
 (= (fin res4) 29)
 (= (personas res5) 1)
 (= (inicio res5) 21)
 (= (fin res5) 28)
 (= (personas res6) 4)
 (= (inicio res6) 10)
 (= (fin res6) 14)
 (= (personas res7) 1)
 (= (inicio res7) 7)
 (= (fin res7) 28)
 (= (personas res8) 1)
 (= (inicio res8) 18)
 (= (fin res8) 25)
 (= (personas res9) 2)
 (= (inicio res9) 27)
 (= (fin res9) 30)
 (= (personas res10) 1)
 (= (inicio res10) 9)
 (= (fin res10) 29)
 (= (personas res11) 3)
 (= (inicio res11) 25)
 (= (fin res11) 29)
 (= (personas res12) 1)
 (= (inicio res12) 7)
 (= (fin res12) 19)
 (= (personas res13) 3)
 (= (inicio res13) 15)
 (= (fin res13) 26)
 (= (personas res14) 3)
 (= (inicio res14) 24)
 (= (fin res14) 27)
 (= (personas res15) 4)
 (= (inicio res15) 26)
 (= (fin res15) 30)
 (= (personas res16) 3)
 (= (inicio res16) 10)

(= (fin res16) 24)
 (= (personas res17) 2)
 (= (inicio res17) 25)
 (= (fin res17) 29)
 (= (personas res18) 4)
 (= (inicio res18) 25)
 (= (fin res18) 30)
 (= (personas res19) 2)
 (= (inicio res19) 27)
 (= (fin res19) 30)
 (= (personas res20) 1)
 (= (inicio res20) 15)
 (= (fin res20) 16)
 (= (personas res21) 2)
 (= (inicio res21) 13)
 (= (fin res21) 23)
 (= (personas res22) 1)
 (= (inicio res22) 25)
 (= (fin res22) 29)
 (= (personas res23) 4)
 (= (inicio res23) 28)
 (= (fin res23) 30)
 (= (personas res24) 4)
 (= (inicio res24) 8)
 (= (fin res24) 22)
 (= (personas res25) 3)
 (= (inicio res25) 18)
 (= (fin res25) 28)
 (= (personas res26) 4)
 (= (inicio res26) 17)
 (= (fin res26) 30)
 (= (personas res27) 1)
 (= (inicio res27) 17)
 (= (fin res27) 19)
 (= (personas res28) 4)
 (= (inicio res28) 21)
 (= (fin res28) 30)
 (= (personas res29) 4)
 (= (inicio res29) 11)
 (= (fin res29) 30)
 (= (personas res30) 3)
 (= (inicio res30) 13)
 (= (fin res30) 14)
 (= (personas res31) 3)
 (= (inicio res31) 25)
 (= (fin res31) 27)
 (= (personas res32) 4)
 (= (inicio res32) 3)
 (= (fin res32) 13)
 (= (personas res33) 3)
 (= (inicio res33) 7)
 (= (fin res33) 20)
 (= (personas res34) 4)

```
(= (inicio res34) 14)
(= (fin res34) 17)
(= (personas res35) 2)
(= (inicio res35) 18)
(= (fin res35) 21)
)
(:goal
 (forall (?res - reserva) (resuelta?res))
)
(:metric minimize (reservasNoAsignadas))
)
```

B.7.2 Solución

step 0: DENEGAR_RESERVA RES35 HAB7
 1: DENEGAR_RESERVA RES34 HAB7
 2: DENEGAR_RESERVA RES33 HAB7
 3: DENEGAR_RESERVA RES32 HAB7
 4: DENEGAR_RESERVA RES31 HAB7
 5: DENEGAR_RESERVA RES30 HAB7
 6: DENEGAR_RESERVA RES29 HAB7
 7: DENEGAR_RESERVA RES28 HAB7
 8: DENEGAR_RESERVA RES26 HAB7
 9: DENEGAR_RESERVA RES25 HAB7
 10: DENEGAR_RESERVA RES24 HAB7
 11: DENEGAR_RESERVA RES23 HAB7
 12: DENEGAR_RESERVA RES21 HAB7
 13: DENEGAR_RESERVA RES19 HAB7
 14: DENEGAR_RESERVA RES18 HAB7
 15: DENEGAR_RESERVA RES17 HAB7
 16: DENEGAR_RESERVA RES16 HAB7
 17: DENEGAR_RESERVA RES15 HAB7
 18: DENEGAR_RESERVA RES14 HAB7
 19: DENEGAR_RESERVA RES13 HAB7
 20: DENEGAR_RESERVA RES11 HAB7
 21: DENEGAR_RESERVA RES9 HAB7
 22: DENEGAR_RESERVA RES6 HAB7
 23: DENEGAR_RESERVA RES2 HAB7
 24: RESERVAR_HABITACION RES4 HAB1
 25: DENEGAR_RESERVA RES7 HAB1
 26: RESERVAR_HABITACION RES22 HAB7
 27: DENEGAR_RESERVA RES1 HAB7
 28: RESERVAR_HABITACION RES3 HAB6
 29: RESERVAR_HABITACION RES5 HAB2
 30: DENEGAR_RESERVA RES10 HAB2
 31: RESERVAR_HABITACION RES27 HAB1
 32: RESERVAR_HABITACION RES8 HAB3
 33: DENEGAR_RESERVA RES12 HAB3
 34: RESERVAR_HABITACION RES20 HAB1

time spent: 0.05 seconds instantiating 0 easy, 2429 hard action templates
 0.00 seconds reachability analysis, yielding 442 facts and 1467 actions
 0.00 seconds creating final representation with 324 relevant facts, 1 relevant
 fluents
 0.00 seconds computing LNF
 0.01 seconds building connectivity graph
 16.70 seconds searching, evaluating 51763 states, to a max depth of 0
 16.76 seconds total time