Inteligencia artificial
Práctica de planificación

Jia Long Ji Qiu
Shuang Long Ji Qiu
You Wu

Curso 2021/2022 Q1

Índice

1. Descripción del problema

En el enunciado de este problema se pide un sistema capaz de asignar peticiones de reserva que recibe un hotel para sus habitaciones durante un mes.

En un principio, la información que se dispone de una habitación es su identificador y el número de personas que puede alojar, siendo este un valor entre 1 y 4. Una reserva también tiene un identificador, y se sabe además el número de personas en la reserva (también entre 1 y 4) y el periodo de esta, fijado por los días de inicio y final.

La asignación de estas reservas debe cumplir ciertos criterios y restricciones, que son diferentes para cada extensión del problema. A lo largo de este documento se comentará cómo se ha abordado la resolución de cada extensión mediante planificadores.

2. Nivel básico

Para el nivel básico se pide un planificador que sea capaz de proporcionar una asignación de reservas a habitaciones en las que caben las personas de la reserva, pudiendo ser más grandes. En la asignación no debe haber solapamiento en las ocupaciones y si no se puede asignar alguna reserva, no se asigna ninguna.

2.1. Modelado del dominio

A partir del enunciado general del problema se pueden extraer dos tipos de objetos, habitación y reserva, los cuales son necesarios para poder guardar la información de cada tipo en forma de predicados y funciones.

Las funciones planteadas representan las características proporcionadas de una habitación o una reserva. En el caso de las habitaciones es necesario saber su capacidad y el día en el que está libre. Para las reservas, se guarda el número de personas en la reserva, el día de inicio y el día final. Esta representación permite controlar que no se produzca ningún solapamiento y que no se sobrepase la capacidad de personas en las asignaciones.

En cuanto a los predicados, se ha implementado un único predicado que indica si una reserva ha sido asignada (servida). Este predicado permite poder representar el objetivo del problema, además de permitir al planificador dejar de tener en cuenta una reserva si ya está bien asignada.

La única acción que se usa es la de asignar una reserva a una habitación, por lo que los parámetros que recibe son un objeto de tipo reserva y otro de tipo habitación. En la precondición se comprueba que la reserva no está asignada aún, que el número de personas en la reserva no sobrepasa la capacidad de la habitación, y que el día de inicio de la reserva sea igual o posterior que el día en el que la habitación está libre. Como consecuencia a haber asignado una reserva a una habitación, el día libre de dicha habitación ha de pasar a ser el día siguiente al día final de la reserva, y la reserva queda como asignada.

2.2. Modelado del problema

Teniendo claro cómo es el dominio modelado, en el problema a resolver se plantea un estado inicial que consiste en unas habitaciones con sus respectivas capacidades, todas con el primer día del mes como día libre, y unas reservas con sus correspondientes características: número de personas de la reserva, día de inicio y día final.

El estado final, es decir, el objetivo del problema, es que todas las reservas hayan sido asignadas. Este objetivo cumple con el requerido en la resolución del nivel básico, ya que el planificador o bien encuentra una asignación para cada reserva, o bien termina la ejecución sin haber encontrado ninguna solución en la que todas las reservas hayan sido asignadas y, por lo tanto, no se asigna ninguna.

2.3. Experimentación

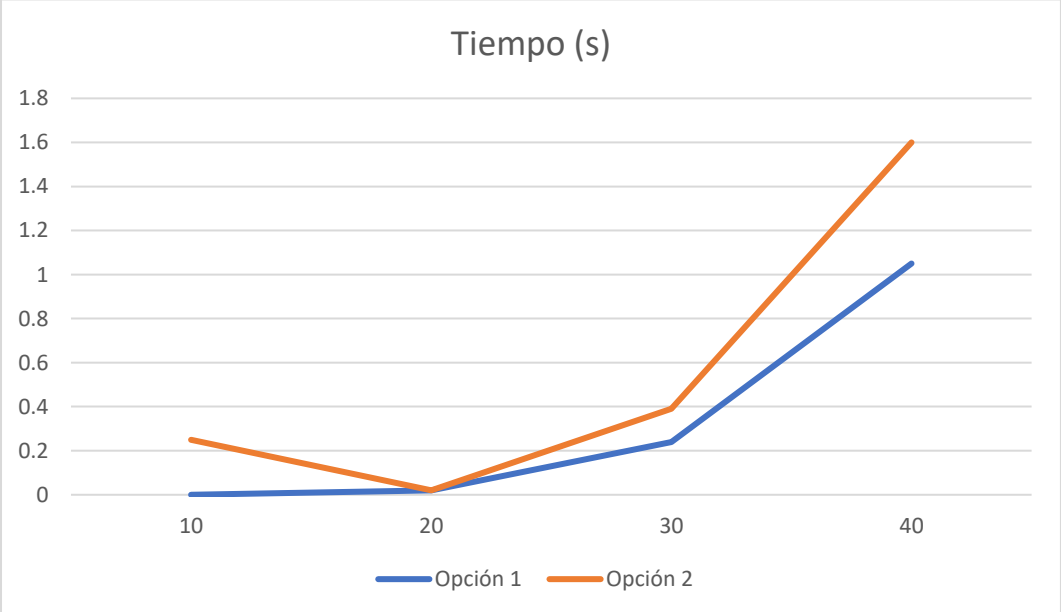
Antes de pasar a explicar los juegos de prueba utilizados para verificar el correcto funcionamiento del planificador, cabe mencionar que para la representación del dominio se han tenido en cuenta dos opciones para resolver el problema del solapamiento.

La primera opción es la comentada anteriormente, donde se utiliza una función que especifica el día libre de una habitación, de manera que una reserva se puede asignar solo si su día de inicio es igual o posterior al día libre de la habitación y, si es asignada, el día libre de la habitación pasa a ser el día posterior al día final de la reserva.

La segunda, en lugar de considerar el día libre de una habitación, a la hora de asignar una reserva se comprueba directamente si las reservas ya asignadas a esa habitación provocan un solapamiento con la reserva que se quiere asignar. Para ello se utiliza un predicado que indica a qué habitación se ha asignado una reserva.

Para decidir qué opción implementar finalmente, se han hecho pruebas con problemas que varían tanto en número de habitaciones como en número de reservas, y se ha tenido en cuenta el número de estados evaluados y el tiempo de ejecución. Los resultados han sido los siguientes:

		Opción 1		Opción 2	
Habitaciones	Reservas	Estados	Tiempo (s)	Estados	Tiempo (s)
10	10	243	0.00	101300	0.25
20	20	2316	0.02	2413	0.02
30	30	8967	0.24	9713	0.39
40	40	19129	1.05	19718	1.60
50	50	36689	3.24	∞	∞



Se puede observar que la primera opción evalúa menos estados y tarda menos en encontrar una solución que la segunda. Además, el planificador modelado según la segunda opción llega a un punto en el que es incapaz de hallar una solución en un intervalo de tiempo apropiado. Por lo tanto, se ha optado por implementar la primera opción.

2.3.1. JP1

En el Juego de Pruebas 1, se ha planteado un problema resoluble en el que se estudia que la solución dada por el planificador realmente cumple con las restricciones requeridas para el nivel básico.

Para ello, se han definido las siguientes habitaciones:

- H1: habitación con capacidad 1, día libre 1
- H2: habitación con capacidad 2, día libre 1
- H3: habitación con capacidad 3, día libre 1
- H4: habitación con capacidad 4, día libre 1

Y en el caso de las reservas:

- R1: reserva para 4 personas, del día 1 hasta el día 7
- R2: reserva para 3 personas, del día 1 hasta el día 15
- R3: reserva para 1 persona, del día 6 hasta el día 10
- R4: reserva para 2 personas, del día 23 hasta el día 29
- R5: reserva para 4 personas, del día 15 hasta el día 20
- R6: reserva para 1 persona, del día 8 hasta el día 14
- R7: reserva para 3 personas, del día 28 hasta el día 30
- R8: reserva para 2 personas, del día 14 hasta el día 16

El resultado obtenido ha sido:

- Asignar la reserva R3 a H1
- Asignar la reserva R8 a H2
- Asignar la reserva R4 a H2
- Asignar la reserva R2 a H3
- Asignar la reserva R7 a H3
- Asignar la reserva R1 a H4
- Asignar la reserva R6 a H4
- Asignar la reserva R5 a H4

Esta solución es efectivamente una solución válida. Todas las reservas se han asignado en habitaciones donde la capacidad es suficiente, y además no se produce ningún solapamiento en la ocupación. Por ejemplo, en el caso de las reservas R3 y R6, se puede observar que ambas son para 1 persona y sus periodos de ocupación se solapan, por lo que una de ellas, en este caso ha sido R6, se ha tenido que asignar a otra habitación donde no se produjera dicho solapamiento.

2.3.2. JP2

Para el Juego de Pruebas 2, se ha forzado un problema que no se puede resolver por una única reserva debido a la falta de una habitación con suficiente capacidad.

Para ello, se han definido las siguientes habitaciones:

- H1: habitación con capacidad 1, día libre 1
- H2: habitación con capacidad 2, día libre 1
- H3: habitación con capacidad 3, día libre 1

Y en el caso de las reservas:

- R1: reserva para 4 personas, del día 1 hasta el día 1
- R2: reserva para 3 personas, del día 1 hasta el día 1
- R3: reserva para 1 persona, del día 1 hasta el día 1
- R4: reserva para 2 personas, del día 1 hasta el día 1

El resultado obtenido ha sido el esperado, que el problema planteado no tiene solución, y se puede ver claramente que es debido a la carencia de una habitación con una capacidad para 4 personas, necesaria para la reserva R1.

2.3.3. JP3

Para el Juego de Pruebas 3, también se ha planteado un problema irresoluble, pero esta vez debido al solapamiento.

Para ello, se ha definido una sola habitación:

- H1: habitación con capacidad 1, día libre 1

Y en el caso de las reservas:

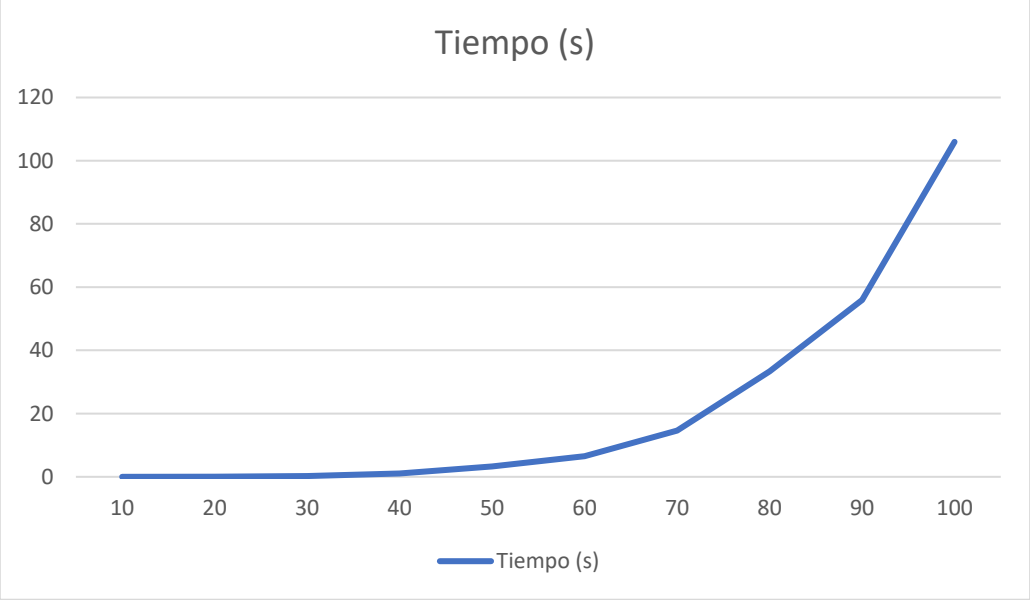
- R1: reserva para 1 personas, del día 1 hasta el día 8
- R2: reserva para 1 personas, del día 9 hasta el día 15
- R3: reserva para 1 persona, del día 16 hasta el día 30
- R4: reserva para 1 personas, del día 4 hasta el día 4

Una vez más, el resultado obtenido ha sido que el problema planteado no tiene solución, y esta vez no por la falta de habitaciones con una capacidad suficiente, sino porque hay solapamiento entre las reservas R1 y R4.

2.3.4. Influencia del tamaño del problema en el tiempo de resolución

Para estudiar la influencia del tamaño del problema en el tiempo de resolución, se ha partido de 10 habitaciones y 10 reservas, y se han ido incrementando de 10 en 10. Los resultados obtenidos han sido los siguientes:

Habitaciones	Reservas	Estados	Tiempo (s)
10	10	243	0.00
20	20	2316	0.02
30	30	8967	0.24
40	40	19129	1.05
50	50	36689	3.24
60	60	54157	6.49
70	70	87456	14.61
80	80	141218	33.37
90	90	184245	55.95
100	100	254492	105.97



Se puede observar que el tiempo de resolución crece exponencialmente según incrementa el número de habitaciones y reservas.

3. Extensión 1

En esta extensión se pide un planificador que sea capaz de proporcionar una asignación de reservas a habitaciones en las que caben un número de personas, pudiendo ser más grandes. En la asignación no puede haber ningún solapamiento en las ocupaciones y el objetivo es maximizar el número de reservas asignadas, teniendo en cuenta que pueden quedar reservas sin asignar.

3.1. Modelado del dominio

Partiendo de la implementación del nivel básico como base, en esta primera extensión se ha añadido una función de reservas descartadas y una acción para descartar reservas. Además, servir una reserva ahora hace referencia a que haya sido asignada o descartada.

La función de reservas descartadas permite al modelado del problema utilizarla como criterio a ser minimizado.

La acción de descartar una reserva recibe como parámetro una reserva. En la precondición se comprueba que la reserva no esté servida aún y que no se pueda asignar a ninguna habitación, ya sea por una capacidad insuficiente o bien el día libre es posterior al día de inicio de la reserva. Con esta precondición se garantiza que una reserva realmente ha quedado inasignable. Como consecuencia de haber descartado una reserva, la reserva queda como servida y se incrementa en uno el número de reservas descartadas.

3.2. Modelado del problema

En cuanto al modelado del problema, además de lo explicado en el nivel básico, en el estado inicial se ha añadido la función de reservas descartadas inicializada a 0, y se ha definido una métrica para minimizar el número de reservas descartadas, la cual depende de la función de reservas asignadas.

3.3. Experimentación

Para la experimentación de la extensión 1, se han utilizado los mismos juegos de pruebas del nivel básico, haciendo los cambios necesarios para adaptarlo a esta extensión. Esta vez el objetivo ha sido, además de comprobar su correcto funcionamiento, verificar que para los juegos de pruebas donde había alguna reserva que no podía asignarse esta vez sí que se encontraba una solución, la cual descarta aquellas reservas que producían conflictos en la asignación.

3.3.1. JP1

Recordemos que en el Juego de Pruebas 1 del nivel básico se habían definido las siguientes habitaciones:

- H1: habitación con capacidad 1, día libre 1
- H2: habitación con capacidad 2, día libre 1
- H3: habitación con capacidad 3, día libre 1
- H4: habitación con capacidad 4, día libre 1

Y en el caso de las reservas:

- R1: reserva para 4 personas, del día 1 hasta el día 7
- R2: reserva para 3 personas, del día 1 hasta el día 15
- R3: reserva para 1 persona, del día 6 hasta el día 10
- R4: reserva para 2 personas, del día 23 hasta el día 29
- R5: reserva para 4 personas, del día 15 hasta el día 20
- R6: reserva para 1 persona, del día 8 hasta el día 14
- R7: reserva para 3 personas, del día 28 hasta el día 30
- R8: reserva para 2 personas, del día 14 hasta el día 16

El resultado obtenido ha sido:

- Asignar la reserva R3 a H1
- Asignar la reserva R8 a H2
- Asignar la reserva R4 a H2
- Asignar la reserva R2 a H3
- Asignar la reserva R7 a H3
- Asignar la reserva R1 a H4
- Asignar la reserva R6 a H4
- Asignar la reserva R5 a H4

Esta solución es exactamente la misma que se obtuvo con el planificador del nivel básico.

3.3.2. JP2

En el Juego de Pruebas 2, el problema no se podía resolver por una reserva que no se había podido asignar debido a la falta de una habitación con suficiente capacidad.

Las habitaciones eran:

- H1: habitación con capacidad 1, día libre 1
- H2: habitación con capacidad 2, día libre 1
- H3: habitación con capacidad 3, día libre 1

Y las reservas:

- R1: reserva para 4 personas, del día 1 hasta el día 1
- R2: reserva para 3 personas, del día 1 hasta el día 1
- R3: reserva para 1 persona, del día 1 hasta el día 1
- R4: reserva para 2 personas, del día 1 hasta el día 1

El resultado obtenido ha sido el siguiente:

- Descartar la reserva R1
- Asignar la reserva R3 a H1
- Asignar la reserva R4 a H2
- Asignar la reserva R2 a H3

Esta vez se puede observar que la reserva R1, la cual era la reserva que creaba conflicto debido a que se carece de una habitación para 4 personas, se ha descartado y el resto se han asignado correctamente.

3.3.3. JP3

El Juego de Pruebas 3 era irresoluble debido a dos reservas que se solapaban.

Se había definido la siguiente habitación:

- H1: habitación con capacidad 1, día libre 1

Y de reservas había:

- R1: reserva para 1 persona, del día 1 hasta el día 8
- R2: reserva para 1 persona, del día 9 hasta el día 15
- R3: reserva para 1 persona, del día 16 hasta el día 30
- R4: reserva para 1 persona, del día 4 hasta el día 4

El resultado obtenido ha sido el siguiente:

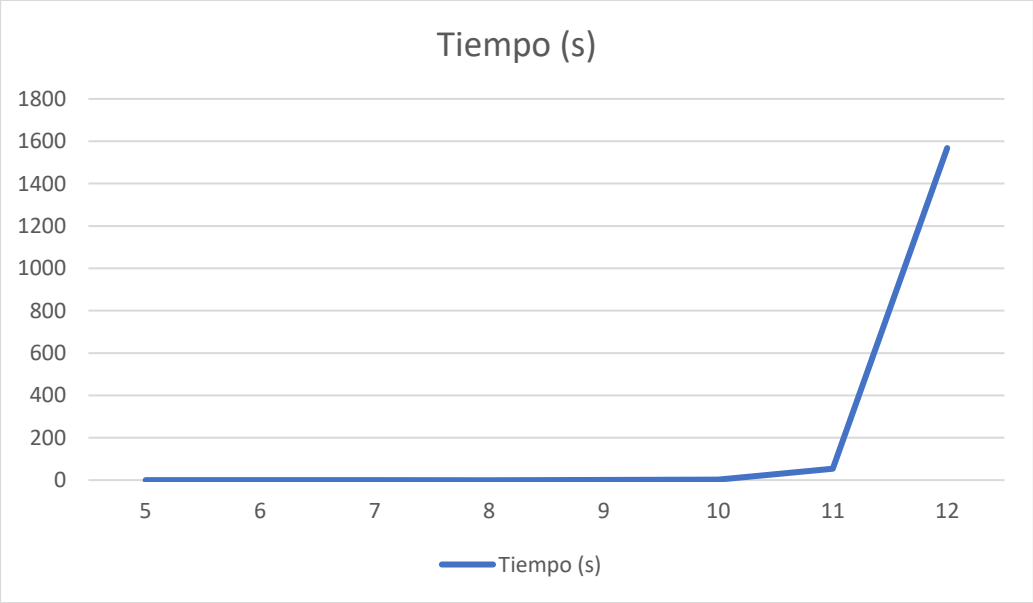
- Asignar la reserva R1 a H1
- Descartar la reserva R4
- Asignar la reserva R2 a H1
- Asignar la reserva R3 a H1

En este caso las reservas que creaban conflicto debido al solapamiento eran R1 y R4. Dos soluciones válidas serían o bien descartar R1 o bien R4, y el planificador ha optado por descartar R4 debido al orden en el que se han definido los objetos.

3.3.4. Influencia del tamaño del problema en el tiempo de resolución

Para estudiar la influencia del tamaño del problema en el tiempo de resolución, se ha fijado el número de habitaciones a 5 y se ha ido incrementando el número de reservas de 1 en 1, partiendo de 5. Los resultados obtenidos han sido los siguientes:

Habitaciones	Reservas	Estados	Tiempo (s)
5	5	45	0.00
5	6	56	0.00
5	7	190	0.00
5	8	5864	0.02
5	9	14795	0.20
5	10	51499	2.89
5	11	178825	53.64
5	12	539767	1568.10



La extensión 1, a diferencia del nivel básico, crece exponencialmente con cambios muy pequeños en el tamaño del problema. Esto se debe a que para esta extensión ya no se tiene una única acción, y por lo tanto el tamaño de búsqueda también crece.

4. Extensión 2

En esta extensión en las reservas se puede indicar una orientación preferente de la habitación. El objetivo es obtener una asignación que maximice el número de reservas asignadas a una habitación con la orientación solicitada, pero teniendo en cuenta que es peor no asignar una reserva que asignarle una orientación incorrecta.

4.1. Modelado del dominio

Partiendo de la extensión 1, se han añadido dos nuevas funciones relacionadas con los objetos habitación y reserva. La primera función indica la orientación en la cual se encuentra la habitación y la segunda indica la orientación preferente de la reserva. Para simplificar, cada orientación se representa con un valor numérico. También se han implementado otra función que sirve para guardar cuántas reservas tienen una orientación incorrecta.

En cuanto a los acciones, se ha añadido una nueva acción que asigna reservas con preferencia de orientación y también se ha modificado el efecto de la acción asignar sin ninguna preferencia.

La acción de asignar con preferencia recibe como parámetro una reserva y una habitación. Como precondición se verifica que la reserva aún no está servida, que pueda ocupar la habitación y finalmente que la orientación que se prefiere de la reserva coincide con la orientación de la habitación. En el caso que se cumpla la precondición, la reserva se indicará como servida y se actualizará el día libre de la habitación.

El efecto de la acción de asignar sin ninguna preferencia ahora también incrementa la función de reservas asignadas con una orientación incorrecta.

4.2. Modelado del problema

En el estado inicial ahora se indica la orientación de una habitación y la orientación preferente de una reserva. Las orientaciones N, S, E y O se representan con 1, 2, 3 y 4, respectivamente. La función de reservas con una orientación incorrecta se inicializa a 0.

El objetivo del problema se mantiene igual al de la extensión 1, pero esta vez la función métrica a minimizar es el siguiente:

$$(+\ (*\ 10\ (reservas - descartadas))\ (reservas - orientacion - incorrecta))$$

Estos factores de escalado se han asignado según el criterio de resolución de esta segunda extensión. Sabiendo que es peor no asignar una reserva que asignarla con una orientación incorrecta, la penalización que supone descartar una reserva es mucho mayor que asignar una reserva con una orientación incorrecta. De este modo, la prioridad que da el planificador a las acciones es, en orden, asignar una reserva con una orientación correcta, asignar una reserva con una orientación incorrecta y descartar una reserva.

4.3. Experimentación

Para la experimentación de la extensión 2, el objetivo ha sido verificar que, además de que se cumpla el criterio de la extensión 1, se le da preferencia a aquellas asignaciones que pueden cumplir con la orientación preferente de la reserva.

4.3.1. JP1

En el Juego de Pruebas 1, se ha planteado un problema donde de todas las reservas que hay, solo se puede asignar una. El objetivo sería entonces priorizar la reserva cuya orientación coincide con la habitación disponible, y el resto se descartaría.

Para ello, se ha definido la siguiente habitación:

- H1: habitación con capacidad 1, día libre 1, orientada hacia el N

Y de reservas tenemos:

- R1: reserva para 1 persona, del día 1 hasta el día 1, con orientación preferente S
- R2: reserva para 1 persona, del día 1 hasta el día 1, con orientación preferente N
- R3: reserva para 1 persona, del día 1 hasta el día 1, con orientación preferente E
- R4: reserva para 1 persona, del día 1 hasta el día 1, con orientación preferente O

El resultado obtenido ha sido:

- Asignar la reserva R2 a la habitación H1, la cual tiene una orientación correcta
- Descartar la reserva R1
- Descartar la reserva R3
- Descartar la reserva R4

Se puede observar que, en la solución dada, la reserva asignada ha sido la que efectivamente cumple con la orientación preferente.

4.3.2. JP2

Para el Juego de Pruebas 2, se ha generado un problema aleatorio con 5 habitaciones y 5 reservas y se ha comprobado si realmente se encuentra la solución óptima.

Las habitaciones generadas han sido:

- H1: habitación con capacidad 3, día libre 1, orientada hacia el O
- H2: habitación con capacidad 4, día libre 1, orientada hacia el S
- H3: habitación con capacidad 4, día libre 1, orientada hacia el S
- H4: habitación con capacidad 1, día libre 1, orientada hacia el E
- H5: habitación con capacidad 2, día libre 1, orientada hacia el S

Y en el caso de las reservas:

- R1: reserva para 4 personas, del día 3 hasta el día 7, con orientación preferente E
- R2: reserva para 3 personas, del día 24 hasta el día 28, con orientación preferente N
- R3: reserva para 1 persona, del día 23 hasta el día 25, con orientación preferente O
- R4: reserva para 3 personas, del día 28 hasta el día 29, con orientación preferente E
- R5: reserva para 4 personas, del día 3 hasta el día 25, con orientación preferente O

El resultado obtenido ha sido el siguiente:

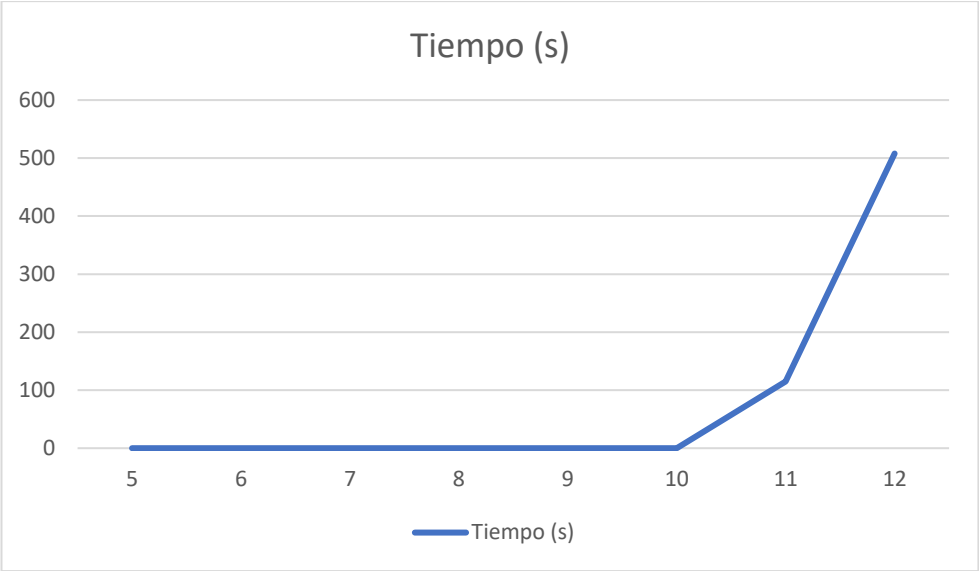
- Asignar la reserva R1 a la habitación H2
- Asignar la reserva R2 a la habitación H2
- Asignar la reserva R5 a la habitación H3
- Asignar la reserva R4 a la habitación H3
- Asignar la reserva R3 a la habitación H1, la cual tiene una orientación correcta

En este problema, la única reserva que se puede asignar con una orientación correcta es la reserva R3 a la habitación H1. El resto de las reservas que potencialmente se podrían asignar con una orientación correcta, no se pueden asignar debido a la falta de capacidad en las habitaciones. Por lo tanto, la solución proporcionada por el planificador es correcta.

4.3.3. Influencia del tamaño del problema en el tiempo de resolución

Para estudiar la influencia del tamaño del problema en el tiempo de resolución, se ha fijado el número de habitaciones a 5 y se ha ido incrementando el número de reservas de 1 en 1, partiendo de 5. Los resultados obtenidos han sido los siguientes:

Habitaciones	Reservas	Estados	Tiempo (s)
5	5	48	0.00
5	6	65	0.00
5	7	92	0.00
5	8	114	0.00
5	9	151	0.00
5	10	443	0.00
5	11	129023	114.82
5	12	214243	507.87



La extensión 2, a pesar de tener algo más de complejidad que la extensión 1, el tiempo de ejecución no crece tan rápido a medida que se aumenta el tamaño del problema.

5. Extensión 3

En esta extensión el objetivo es maximizar el número de reservas asignadas minimizando el desperdicio de plazas.

5.1. Modelado del dominio

Partiendo de la extensión 1, se ha añadido una función que representa el número de plazas desperdiciadas y se ha modificado el efecto de asignar una reserva a una habitación.

La función de plazas desperdiciadas permite al modelado del problema utilizarla como criterio a ser minimizado.

Como efecto de asignar una reserva a una habitación, ahora también incrementa la función de plazas desperdiciadas según la diferencia entre la capacidad de la habitación y el número de personas de la reserva.

5.2. Modelado del problema

En el estado inicial, además de la implementación de la extensión 1, se inicializa la función de plazas desperdiciadas a 0.

El objetivo del problema se mantiene igual, pero esta vez la función métrica a minimizar es el siguiente:

$$(+ (* \ 10 \ (reservas - descartadas)) \ (plazas - desperdiciadas))$$

Utilizando estos factores de escalado, se da más importancia a minimizar el número de reservas descartadas, pero esta vez también se busca minimizar el número de plazas desperdiciadas, siendo este último criterio menos importante.

5.3. Experimentación

Para esta extensión, se ha comprobado que las asignaciones dadas por el planificador cumplen con el criterio de minimizar las plazas desperdiciadas.

5.3.1. JP1

En el Juego de Pruebas 1, se ha planteado un problema donde de todas las reservas que hay, solo se puede asignar una. El objetivo sería entonces asignar la reserva que menos plazas desperdicie, y el resto se descartaría.

Para ello, se ha planteado una única habitación:

- H1: habitación con capacidad 4, día libre 1

Y de reservas:

- R1: reserva para 2 personas, del día 1 hasta el día 1

- R2: reserva para 4 personas, del día 1 hasta el día 1
- R3: reserva para 3 personas, del día 1 hasta el día 1
- R4: reserva para 1 persona, del día 1 hasta el día 1

El resultado obtenido ha sido el siguiente:

- Asignar la reserva R2 a la habitación H1
- Descartar la reserva R1
- Descartar la reserva R3
- Descartar la reserva R4

Observando el resultado se puede comprobar fácilmente que la reserva que ha sido asignada es la que menos desperdicio de plazas genera, ya que el número de personas de la reserva R2 coincide con la capacidad de la habitación H1.

5.3.2. JP2

Para el Juego de Pruebas 2, se ha generado un problema aleatorio con 5 habitaciones y 5 reservas y se ha comprobado si realmente se encuentra la solución óptima.

Las habitaciones generadas han sido:

- H1: habitación con capacidad 4, día libre 1
- H2: habitación con capacidad 3, día libre 1
- H3: habitación con capacidad 2, día libre 1
- H4: habitación con capacidad 4, día libre 1
- H5: habitación con capacidad 2, día libre 1

Y en el caso de las reservas:

- R1: reserva para 1 persona, del día 26 hasta el día 28
- R2: reserva para 3 personas, del día 10 hasta el día 11
- R3: reserva para 4 personas del día 8 hasta el día 24
- R4: reserva para 1 persona, del día 24 hasta el día 28
- R5: reserva para 1 persona, del día 7 hasta el día 15

El resultado obtenido ha sido el siguiente:

- Asignar la reserva R1 a la habitación H4
- Asignar la reserva R4 a la habitación H3
- Asignar la reserva R5 a la habitación H5
- Asignar la reserva R3 a la habitación H1
- Asignar la reserva R2 a la habitación H2

Esta solución, a pesar de ser válida porque cumple con el objetivo del problema, no es la óptima. Se puede observar que la reserva R1, siendo para 1 persona, se ha asignado a la habitación H4, la cual tiene una capacidad para 4 personas, cuando se podría haber asignado a la habitación H5, con capacidad 2, reduciendo así las plazas desperdiciadas.

Tras haber hecho varias ejecuciones sobre el mismo juego de pruebas, pero variando el peso de la función heurística, se ha llegado a la conclusión de que, tal y como está definida la función de evaluación por defecto, $f(n) = g(n) + 5 * h(n)$, se sobreestima el valor de la función heurística y por lo tanto se incumple la propiedad de admisibilidad, razón por la cual no se encuentra la solución óptima.

Reduciendo el peso de la función heurística a 1, el resultado obtenido ha sido el siguiente:

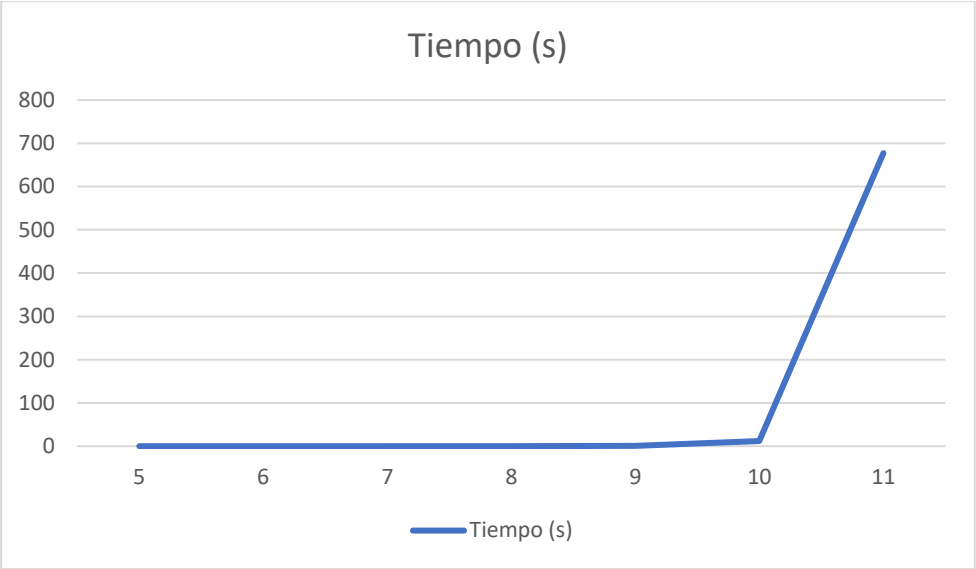
- Asignar la reserva R2 a la habitación H2
- Asignar la reserva R3 a la habitación H1
- Asignar la reserva R1 a la habitación H3
- Asignar la reserva R5 a la habitación H5
- Asignar la reserva R4 a la habitación H5

Esta vez, la solución sí que es óptima.

5.3.3. Influencia del tamaño del problema en el tiempo de resolución

Para estudiar la influencia del tamaño del problema en el tiempo de resolución, se ha fijado el número de habitaciones a 5 y se ha ido incrementando el número de reservas de 1 en 1, partiendo de 5. Los resultados obtenidos han sido los siguientes:

Habitaciones	Reservas	Estados	Tiempo (s)
5	5	42	0.00
5	6	303	0.00
5	7	358	0.00
5	8	6213	0.04
5	9	16757	0.46
5	10	64250	11.54
5	11	243016	677.44
5	12	∞	∞



Se puede observar que el tiempo de resolución crece exponencialmente con cambios muy pequeños en el tamaño del problema. Esta vez, la ejecución para 5 habitaciones y 12 reservas no ha dado ningún resultado dentro de un intervalo de tiempo aceptable.

6. Extensión 4

En esta extensión se pide ampliar la extensión 3 añadiendo un criterio que consiste en minimizar el número de habitaciones diferentes que se ocupan durante un mes. En este caso se han de considerar los pesos de tres criterios, los cuales son, en orden de prioridad (de mayor a menor), minimizar el número de reservas descartadas, minimizar el número de habitaciones diferentes utilizadas y minimizar el número de plazas desperdiciadas.

6.1. Modelado del dominio

Partiendo de la extensión 3, se ha añadido una función que representa el número de habitaciones diferentes utilizadas, además de un nuevo predicado que indica si una habitación ha sido utilizada. En cuanto a las acciones, en esta extensión la acción de asignar una reserva a una habitación ahora se ha dividido en dos casos: asignar una reserva a una habitación nueva y asignar una reserva a una habitación ya utilizada.

Ambas acciones hacen lo mismo que la acción explicada en la extensión 3 en esencia, pero difieren en el tratamiento de las habitaciones.

En el caso de asignar una reserva a una habitación nueva, se comprueba como precondición que la habitación no se haya utilizada, y como efecto se incrementa en 1 la función de habitaciones diferentes y se marca la habitación como utilizada.

Contrariamente, en la acción de asignar una reserva a una habitación ya utilizada, la precondición que se comprueba es que la habitación ya haya sido utilizada. Como efecto no se ha hecho ningún cambio.

6.2. Modelado del problema

En el estado inicial, además de la implementación de la extensión 3, se inicializa a 0 la función de habitaciones diferentes utilizadas.

El objetivo del problema se mantiene igual, pero esta vez la función métrica a minimizar es el siguiente:

$$(+\ (+\ (*\ 100\ (reservas - descartadas))\ (*\ 10\ (habitaciones - diferentes)))\ (plazas - desperdiciadas))$$

Estos tres factores de escaldo se han asignado según el criterio de esta extensión, siendo minimizar el número de reservas descartadas lo más prioritario y minimizar las plazas desperdiciadas lo menos importante.

6.3. Experimentación

Para esta extensión, se ha comprobado que las asignaciones dadas por el planificador cumplen con los tres criterios explicados.

6.3.1. JP1

El Juego de Pruebas 1 se ha diseñado específicamente para verificar que el criterio de minimizar el número de habitaciones diferentes tiene un impacto sobre la solución encontrada.

Para ello, las habitaciones definidas han sido:

- H1: habitación con capacidad 1, día libre 1.
- H2: habitación con capacidad 2, día libre 1.
- H3: habitación con capacidad 3, día libre 1.
- H4: habitación con capacidad 4, día libre 1.

En el caso de las reservas:

- R1: reserva para 4 personas, del día 1 hasta el día 7.
- R2: reserva para 3 personas, del día 1 hasta el día 1.
- R3: reserva para 1 persona, del día 6 hasta el día 10.
- R4: reserva para 2 personas, del día 23 hasta el día 29.
- R5: reserva para 4 personas, del día 15 hasta el día 20.
- R6: reserva para 1 persona, del día 8 hasta el día 13.
- R7: reserva para 3 personas, del día 28 hasta el día 30.
- R8: reserva para 2 personas, del día 14 hasta el día 16.

Un primer resultado se ha obtenido anulando la función que indica el número de habitaciones diferentes en la métrica (asignándole un peso 0), y ha sido el siguiente:

- Asignar la reserva R3 a la habitación nueva H2
- Asignar la reserva R6 a la habitación nueva H1
- Asignar la reserva R8 a la habitación ya utilizada H2
- Asignar la reserva R2 a la habitación nueva H3
- Asignar la reserva R4 a la habitación ya utilizada H2
- Asignar la reserva R7 a la habitación ya utilizada H3
- Asignar la reserva R1 a la habitación nueva H4
- Asignar la reserva R5 a la habitación ya utilizada H4

En esta solución, debido a que no se busca minimizar el número de habitaciones diferentes utilizadas, el criterio que se está optimizando es el número de plazas desperdiciadas.

Reestableciendo el peso de la función de habitaciones diferentes a 10 se ha obtenido la siguiente solución:

- Asignar la reserva R2 a la habitación nueva H3
- Asignar la reserva R1 a la habitación nueva H4

- Asignar la reserva R6 a la habitación ya utilizada H4
- Asignar la reserva R3 a la habitación ya utilizada H3
- Asignar la reserva R4 a la habitación ya utilizada H3
- Asignar la reserva R8 a la habitación nueva H2
- Asignar la reserva R5 a la habitación ya utilizada H4
- Asignar la reserva R7 a la habitación ya utilizada H4

Se puede observar que esta vez el número de habitaciones diferentes utilizadas se ha reducido. Sin embargo, una vez más, esta solución no es óptima debido a que se está sobreestimando la función heurística.

Reduciendo el peso de la función heurística a 1, el resultado obtenido ha sido el siguiente:

- Asignar la reserva R2 a la habitación nueva H3
- Asignar la reserva R1 a la habitación nueva H4
- Asignar la reserva R3 a la habitación ya utilizada H3
- Asignar la reserva R8 a la habitación ya utilizada H3
- Asignar la reserva R4 a la habitación ya utilizada H3
- Asignar la reserva R6 a la habitación ya utilizada H4
- Asignar la reserva R5 a la habitación ya utilizada H4
- Asignar la reserva R7 a la habitación ya utilizada H4

Esta vez, la solución sí es óptima. El número de habitaciones diferentes utilizadas se ha reducido a 2, y el número total de plazas desperdiciadas ha sido 8, siendo esta la mínima debido a que, para las reservas propuestas, no hay otra configuración que permita obtener solo 2 habitaciones diferentes.

6.3.2. JP2

El Juego de Pruebas 2 deriva del 1, y su objetivo es comprobar que, cuando hay un empate de estados en el número mínimo de habitaciones diferentes, se opta por el estado en el que menos desperdicio de plazas hay.

Para ello, el único cambio que se ha hecho respecto al Juego de Pruebas 1 ha sido cambiar el día de inicio de la reserva R1, que ha pasado de ser 1 a ser 2.

El resultado obtenido, reduciendo el peso de la función heurística a 1, ha sido:

- Asignar la reserva R2 a la habitación nueva H4
- Asignar la reserva R3 a la habitación nueva H2
- Asignar la reserva R8 a la habitación ya utilizada H2
- Asignar la reserva R4 a la habitación ya utilizada H2

- Asignar la reserva R1 a la habitación ya utilizada H4
- Asignar la reserva R6 a la habitación ya utilizada H4
- Asignar la reserva R5 a la habitación ya utilizada H4
- Asignar la reserva R7 a la habitación ya utilizada H4

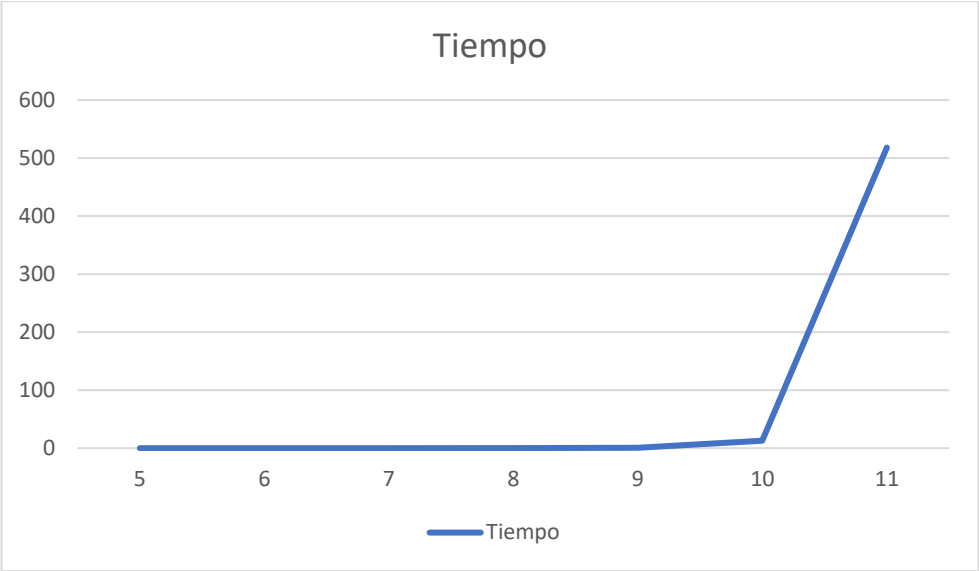
En esta solución, el número de habitaciones diferentes utilizadas ha sido 2, y el desperdicio de plazas ha sido 6.

Recordemos que el único cambio hecho respecto al juego de pruebas anterior ha sido reducir el periodo de la reserva R1, por lo que la solución óptima que se podría haber dado en este caso podría haber sido el mismo, donde todas las reservas se asignan entre las habitaciones H3 y H4. Sin embargo, entre la solución anterior y la que se ha hallado en este juego de pruebas, se ha podido reducir el desperdicio de plazas moviendo la reserva R2 a H4 (ahora que ya no solapa con la reserva R1), y el resto de las reservas asignadas a la habitación H3 a la habitación H2, pues se habían asignado a la habitación H3 únicamente para poder minimizar el número de habitaciones distintas utilizadas, a pesar de que se podrían haber asignado perfectamente a la habitación H2.

6.3.3. Influencia del tamaño del problema en el tiempo de resolución

Para estudiar la influencia del tamaño del problema en el tiempo de resolución, se ha fijado el número de habitaciones a 5 y se ha ido incrementando el número de reservas de 1 en 1, partiendo de 5. Los resultados obtenidos han sido los siguientes:

Habitaciones	Reservas	Estados	Tiempo (s)
5	5	60	0.00
5	6	69	0.00
5	7	86	0.00
5	8	6194	0.06
5	9	16750	0.65
5	10	64243	12.89
5	11	243595	518.20
5	12	∞	∞



Se puede observar que el tiempo de resolución crece exponencialmente con cambios muy pequeños en el tamaño del problema. Igual que en la extensión 3, la ejecución de 5 habitaciones y 12 reservas no ha podido dar una solución en un intervalo de tiempo aceptable.

7. Conclusiones

A lo largo de esta práctica se ha estudiado cómo funciona la planificación mediante Metric Fast Forward, empleando el lenguaje PDDL.

Se ha observado que el modelado de un problema y su dominio tiene un gran impacto en su resolución, ya sea en la solución hallada o el tiempo de resolución. Por esta razón, a pesar de que a priori la implementación que se ha hecho durante esta práctica parece ser funcional y correcta, es muy probable que haya otras maneras de plantear el problema que no se han tenido en cuenta y puedan dar mejores resultados.

Respecto a la experimentación sobre cómo influye el tamaño de un problema al tiempo de resolución, ha sido evidente que, cuanta más complejidad había en el modelado del dominio de un problema, o cuanto más grande era el tamaño de un problema, el tiempo de resolución incrementaba exponencialmente.

Finalmente, sobre el lenguaje PDDL cabe mencionar que, al ser un lenguaje declarativo, en muchas ocasiones ha sido difícil saber el motivo por el que algunos resultados no eran los esperados, y se ha requerido hacer cambios por iteraciones en el modelado para solucionarlo.

8. Planificación

8.1. Primera semana (20/12)

Durante la primera semana se ha hecho una primera toma de contacto con el lenguaje PDDL y se ha modelado el nivel básico, además de su generador de problemas.

8.2 Segunda semana (27/12)

En la segunda semana se han implementado todas las extensiones del problema, cada una con su generador correspondiente.

También se ha terminado la documentación de la práctica, que incluye la explicación y la experimentación de cada extensión.