

Heurística y Optimización

GRADO EN INGENIERÍA INFORMÁTICA

Práctica 2

Curso 2020/2021

Jorge Rodríguez Fraile, 100405951, Grupo 81, 100405951@alumnos.uc3m.es
Carlos Rubio Olivares, 100405834, Grupo 81, 100405834@alumnos.uc3m.es

Índice

ÍNDICE

INTRODUCCIÓN A LOS CONTENIDOS	2
PARTE 1	2
DESCRIPCIÓN DEL MODELO	2
Problema:	2
Variables:	2
Dominio:	2
Restricciones:	2
PARTE 2	4
DESCRIPCIÓN DEL MODELO	4
Estado:	4
Operaciones:	4
Heurísticas:	5
Algoritmo:	6
PARTE 3	6
ANÁLISIS DE LOS RESULTADOS PARTE 1	6
ANÁLISIS DE LOS RESULTADOS PARTE 2	8
bruta:	10
lessOBS:	11
distanciaUbeda:	13
Conclusiones de las pruebas	15
HERRAMIENTAS USADAS	16
CONCLUSIONES	16

INTRODUCCIÓN A LOS CONTENIDOS

En esta práctica vamos a estudiar la satisfacción de problemas con restricciones y la búsqueda informada en dos problemas relacionados con satélites y su ubicación. Esta vez no tendremos que fusionar los dos problemas, y ambos serán codificados en python. Ninguno de los integrantes del grupo tiene un amplio conocimiento de este lenguaje, por lo que llevará un tiempo poder acostumbrarse, aunque esa es la barrera más grande que tendremos que superar, ya que en cuanto a los problemas en sí creemos que, si le prestamos la suficiente atención a sus enunciados y utilizamos los apuntes y conocimientos de clase, podremos resolverlos sin mayor dificultad.

PARTE 1

DESCRIPCIÓN DEL MODELO

Se ha diseñado un modelo de restricciones para satisfacer las restricciones de asignación de antenas de transmisión a satélites teniendo en cuenta las franjas horarias por las que pasan sobre las distintas antenas. Donde todas las antenas deberán tener un satélite asignado para su transmisión. Para el diseño del siguiente modelado se ha tenido en cuenta la siguiente tabla:

Satélite	Franja	Antenas
SAT1	00:00 - 12:00	ANT1, ANT2, ANT3, ANT4
SAT2	00:00 - 12:00	ANT1, ANT2, ANT3
SAT3	06:00 - 12:00	ANT4, ANT6
SAT3	13:00 - 16:00	ANT7, ANT9, ANT10
SAT4	16:00 - 00:00	ANT8, ANT11, ANT12
SAT5	06:00 - 13:00	ANT1, ANT7, ANT12
SAT6	09:00 - 13:00	ANT7, ANT9
SAT6	13:00 - 19:00	ANT3, ANT4, ANT5

Problema:

$$P = (X, D, C)$$

Variables:

$$X = \{SAT1, SAT2, SAT31, SAT32, SAT4, SAT5, SAT61, SAT62\}$$

Cada una de las variables representa una franja de tiempo, de esta manera podemos asignar a cada una de las franjas una antena. Por lo tanto, SAT1 es la franja 1 en la que está el satélite 1 y SAT32 es la cuarta franja, pero la segunda para el satélite 3.

Dominio:

$$D = \{D_{SAT1}: \{1, 2, 3, 4\}, D_{SAT2}: \{1, 2, 3\}, D_{SAT31}: \{4, 6\}, D_{SAT32}: \{7, 9, 10\}, D_{SAT4}: \{8, 11, 12\}, \\ D_{SAT5}: \{1, 7, 12\}, D_{SAT61}: \{7, 9\}, D_{SAT62}: \{3, 4, 5\}\}$$

Según la tabla proporcionada, cada una de las franjas puede tomar esas antenas, las antenas se han representado con su valor numérico, es decir ANT1 es 1.

Restricciones:

$$C = \{R_{SAT1, SAT2}, R_{SAT2, SAT4}, R_{SAT2, SAT5}, R_{SAT4, SAT5}, R_{SAT5, SAT4}, R_{SAT32, SAT4}, R_{SAT5, SAT61}\} \\ R_{SAT1, SAT2} = \{(1, 1), (2, 2), (3, 3)\}$$

El satélite 1 y 2 deben estar asignados a la misma antena, por lo que solo pueden tomar el mismo valor y quedan excluidos aquellos valores del dominio que no tienen en común.

$$R_{SAT2,SAT4} = \{(1,8), (1,11), (1,12), (2,8), (2,11), (2,12), (3,8), (3,11), (3,12)\}$$

$$R_{SAT2,SAT5} = \{(1,7), (1,12), (2,1), (2,7), (2,12), (3,1), (3,7), (3,12)\}$$

$$R_{SAT4,SAT5} = \{(8,1), (8,7), (8,12), (11,1), (11,7), (11,12), (12,1)\}$$

Los satélites 2, 4 y 5 no pueden estar asignados a los mismos satélites, por lo que se contempla por pares los posibles valores que pueden tomar.

$$R_{SAT5,SAT4} = \{(12,8), (12,12), (1,8), (1,11), (1,12), (7,8), (7,11)\}$$

Se contemplan todas las combinaciones de los dominios del satélite 4 y 5, excluyendo la asignación SAT5 ANT12 con SAT4 ANT11.

$$R_{SAT32,SAT4} = \{(7,8), (7,11), (7,12), (9,8), (9,11), (9,12), (10,8), (10,11), (10,12)\}$$

$$R_{SAT32,SAT5} = \{(7,1), (7,7), (9,1), (9,7), (9,12), (10,1), (10,7), (10,12)\}$$

$$R_{SAT61,SAT4} = \{(7,8), (7,11), (9,8), (9,11), (9,12)\}$$

$$R_{SAT5,SAT61} = \{(1,7), (1,9), (7,7), (7,9), (12,7), (12,9)\}$$

La franja 32 es de tarde y la 4 también, ambas tienen las antenas 7 y 12, por lo que sería una combinación válida. En segundo lugar, tenemos las franjas 4 y 5, pero como una es de tarde y otra de mañana, no podemos tenerla en cuenta. La siguiente combinación es la franja 32 y 5, pero nos encontramos con el mismo problema que en el caso anterior. La franja 5 con la 61, en cambio, sí que es correcta, ya que ambas franjas son de mañana. Por último, tenemos la franja 4 con la 61, pero no podemos aceptarla, ya que 4 es de tarde y 61 de mañana.

Las restricciones son no direccionales, por lo que da igual el sentido en el que se asigne, que serán válidas.

PARTE 2

DESCRIPCIÓN DEL MODELO

Estado:

Cada estado está compuesto por la información de ambos satélites de esta manera podemos controlar de manera óptima, sus valores están definidos dentro de la clase Nodo. Componentes de cada estado:

satelite.parent: Puntero al nodo padre, que nos posibilita hacer backtracking.
satelite.pos: Indica en qué hora se encuentran ambos satélites, ya que avanzan a la vez, a una hora por profundidad.

satelite.bateria1: Indica la cantidad de batería que tiene el SAT1 para ese estado.

satelite.sat1franjas: Franjas en las que puede observar el SAT1.

satelite.observaciones1: Número de observaciones del SAT1 que todavía no ha transmitido.

satelite.operacion1: Nombre de la operación realizada por el SAT1.

satelite.bateria2: Cantidad de batería que le queda al SAT2.

satelite.sat2franjas: Franjas en las que opera SAT2 en ese momento.

satelite.observaciones: Observaciones realizadas, pero sin transmitir de SAT2.

satelite.operacion2: Nombre de la última operación de SAT2.

satelite.numOBSrestantes: Indica cuantas observaciones de las totales quedan por transmitir. Nos permitirá controlar la condición de salida.

satelite.g: Coste medido en número de operaciones desde el nodo origen.

satelite.h: Coste de la heurística para ese estado.

satelite.f: Suma de f y h, representa el coste total del estado, que nos permite orientar la toma de decisiones del algoritmo.

satelite.OBS: Lista de coordenadas de las observaciones que faltan por observar.

Operaciones:

En cuanto a las operaciones, se diferencian 6: Observar, Girar, IDLE, Carga, Transmisión y Noche.

- Observar: Esta operación consiste en que si cualquiera de los 2 satélites se encuentra en la posición de una observación pueda tomarla, y en ese caso la observación se eliminaría del grid, por tanto, las condiciones que se tienen que dar para poder observar es que se tenga la suficiente batería, el satélite esté en la franja y posición adecuada, y que esa posición no pertenezca al ciclo de noche. Para conocer las coordenadas de las observaciones se han guardado en una lista, donde cada elemento contiene las dos coordenadas de estas.
- Girar: La siguiente operación que se ha tenido en cuenta ha sido girar, se trata, en el caso del primer satélite cambiar su franja de observación de (0,1) a (1,2) y viceversa, mientras que en el caso de satélite 2 de (2,3) a (1,2). Los requerimientos para esto serán más básicos que en el caso anterior, que es que se tenga batería suficiente y

que no nos encontremos en el ciclo de noche. Manejar en qué franja de observación se encuentran los satélites para poder cambiarlas es bastante sencillo ya que guardamos su valor en `nodo.sat1franj` y `nodo.sat2franj` respectivamente.

- IDLE: Esta operación es aquella en la que los satélites no hacen nada, simplemente se mueven a la siguiente posición sin gastar ningún tipo de recurso, por consiguiente, ha sido la operación más fácil de implementar, y sólo tiene el prerequisite de que el satélite no se encuentre en el ciclo de noche.
- Carga: Para esta operación se debe tener en cuenta cuánta batería se tiene, ya que $\text{carga} \times \text{unidades}$ el nivel de batería, donde x viene dado en el fichero de entrada de datos. Por tanto, como máximo debemos tener `maximo_nivel_bateria - recarga_bateria` de unidades para poder realizar la operación, aparte de no estar en el ciclo de noche como todos los casos anteriores.
- Transmisión: La operación de transmisión está muy ligada a la de Observar, ya que sólo si un satélite tiene en su 'stack' una observación puede realizar la operación transmitir, que hará que dicha observación desaparezca de sus datos. Aparte de esta condición tenemos las básicas de que se tenga la batería suficiente y no estemos en el ciclo de noche.
- Noche: La última operación es la única que es 'obligatoria' ya que, si la posición de los satélites pasa a ser 12, significa que han entrado en ciclo de noche, por lo que necesitarán otros 12 turnos para volver al inicio de nuestro 'grid' y realizar las demás operaciones.

Heurísticas:

- Menos observaciones (`lessOBS`): La primera de las heurísticas nace de relajar las precondiciones de posición, franjas y batería. Si los satélites se pueden mover a donde sea en un solo movimiento sin importarle la batería que gasten, el nº de observaciones restantes en el tablero será el factor que diferencia si un movimiento es mejor que otro.
- Distancia Úbeda (`distanciaUbeda`): Esta heurística nace de eliminar las condiciones de batería, franjas, y solo moverse poder moverse en una dirección (derecha, en este caso), esto fuerza a que el movimiento que haga que podamos acercarnos más a una posible observación impere sobre los demás movimientos.
- Fuerza bruta (`bruta`): Esta heurística no aporta ningún tipo de información, sea cual sea el estado en el que nos encontremos. Ha sido creada fundamentalmente para realizar pruebas de rendimiento y poder observar la mejoras del uso de heurísticas informadas.

En cuanto a la admisibilidad de las heurísticas vemos que todas las heurísticas son admisibles para el primer escenario, todas proporcionan un coste heurístico menor al coste de la solución óptima del problema.

`lessOBS`: $3 \leq 8$ `distanciaUbeda`: $7 \leq 8$

Los resultados anteriores son para el escenario dado, donde `distanciaUbeda` da mejores resultados al estar más próximo al coste real, pero esto ocurre en todos los escenarios por lo que no podemos decir que sea admisible para cualquier escenario. Los resultados de esas pruebas están más adelante, en la sección de pruebas.

`lessOBS` siempre encontrará la solución y será la óptima, dado también que el número de sucesores para un nodo son finitos y los coste de operación no son nulos.

Con respecto a cuál de nuestras heurísticas está más informada, distanciaUbeda en el primer escenario era la favorita al estar muy próxima al real, por lo que expandía menos nodos.

Por lo que, aunque distanciaUbeda era la favorita inicialmente, la heurística que mejores resultados arroja para este problema en general será lessOBS, dado que al estar informada siempre será mejor que bruta y en algunos casos que distanciaUbeda.

Algoritmo:

Para hacer posible el algoritmo A* hemos creado una clase nodo para guardar los estados del problema como se han definido anteriormente. Una vez hecho esto, se empieza el problema añadiendo el nodo inicial a nuestra lista de nodos abiertos, teniendo en cuenta el problema pasado como argumento. El siguiente paso es recorrer la lista abierta para expandir un nodo que no se haya expandido ya, es decir que no se encuentre en la lista cerrada, y entonces se añade a la lista cerrada, en caso de que este nodo expandido sea final, no queden observaciones por transmitir, el algoritmo acaba. En caso de que no lo sea, debemos generar sus hijos, viendo todas las posibles combinaciones de operaciones entre los dos satélites, una vez generados y añadidos a la lista abierta, esta última se ordena teniendo en cuenta el valor de $f = g + h$ para poder así expandir el nodo con menos f en la siguiente iteración.

PARTE 3

ANÁLISIS DE LOS RESULTADOS PARTE 1

Se han obtenido 468 posibles soluciones para los datos dados aplicando las restricciones impuestas y explicadas anteriormente. Por ejemplo, una solución válida para este modelo sería:

SAT1 00:00 - 12:00 ANT3
SAT2 00:00 - 12:00 ANT3
SAT3 06:00 - 12:00 ANT4
SAT3 13:00 - 16:00 ANT10
SAT4 16:00 - 00:00 ANT12
SAT5 13:00 - 06:00 ANT1
SAT6 09:00 - 13:00 ANT9
SAT6 13:00 - 19:00 ANT5

Si observamos los resultados obtenidos, podemos ver que se cumplen todas las restricciones que hemos aplicado. Para entender en más profundidad el problema que se nos da, vamos a añadir un nuevo satélite llamado SAT7 que pueda dar cobertura a las antenas ANT6, ANT9 y ANT10 y que este operativo de 13:00 a 16:00.

Con esta nueva combinación salen 1404 soluciones, casi el doble que, en el problema original, ya que añadir un satélite da muchas más opciones de asignación, además hemos evitado cualquier conflicto mayor con las restricciones para ver como de significativa sería el aumento de soluciones. Un ejemplo sería:

SAT1 00:00 - 12:00 ANT2
SAT2 00:00 - 12:00 ANT2
SAT3 06:00 - 12:00 ANT4
SAT3 13:00 - 16:00 ANT9
SAT4 16:00 - 00:00 ANT8
SAT5 13:00 - 06:00 ANT7
SAT6 09:00 - 13:00 ANT7
SAT6 13:00 - 19:00 ANT6
SAT7 13:00 - 16:00 ANT9

La siguiente prueba que hemos realizado ha sido borrar la SAT3.1, en este caso obtenemos 234 soluciones, es algo obvio, ya que el eliminar una variable de asignación limita nuestras posibilidades, sobre todo si es una bastante flexible, como es el caso de esta. En definitiva, el resultado es la mitad de lo que teníamos, un ejemplo de esto es:

SAT1 00:00 - 12:00 ANT2
SAT2 00:00 - 12:00 ANT2
SAT3 06:00 - 12:00 ANT7
SAT4 16:00 - 00:00 ANT11
SAT5 13:00 - 06:00 ANT1
SAT6 09:00 - 13:00 ANT7
SAT6 13:00 - 19:00 ANT5

Otra prueba que hemos querido analizar ha sido añadir la antena, ANT13 con cobertura en las franjas SAT3 06:00 - 12:00 y SAT6 09:00 - 13:00, el resultado ha sido de 1080 soluciones, el resultado es muy similar a la primera prueba que hemos realizado, ya que aportamos más variables que dan lugar a todavía más estados de asignación. Una solución particular en este caso es:

SAT1 00:00 - 12:00 ANT3
SAT2 00:00 - 12:00 ANT3
SAT3 06:00 - 12:00 ANT13
SAT3 13:00 - 16:00 ANT7
SAT4 16:00 - 00:00 ANT8
SAT5 13:00 - 06:00 ANT1
SAT6 09:00 - 13:00 ANT9
SAT6 13:00 - 19:00 ANT5

La última prueba que vamos a realizar es quitar la antena ANT4, esto obliga a la franja 1 de SAT3 a tomar el valor 6, obteniendo 156 soluciones. El resultado limita mucho las posibilidades de asignación, ya que ANT4 aparece muchas veces en los diferentes satélites. Un resultado de esta prueba es:

SAT1 00:00 - 12:00 ANT1
SAT2 00:00 - 12:00 ANT1
SAT3 06:00 - 12:00 ANT6
SAT3 13:00 - 16:00 ANT9
SAT4 16:00 - 00:00 ANT11
SAT5 13:00 - 06:00 ANT7
SAT6 09:00 - 13:00 ANT7
SAT6 13:00 - 19:00 ANT3

ANÁLISIS DE LOS RESULTADOS PARTE 2

Debemos recalcar que, a la hora de realizar las pruebas, no se ha tenido en cuenta la opción de poder añadir o eliminar un satélite, ya que esto supondría cambiar completamente el diseño del código del problema, y se deberían tener en cuenta nuevas restricciones y movimientos. Por otro lado, para aportar algo más de información en cuanto a la realización de las pruebas, se aportan a continuación las características principales de la máquina en la que se han ejecutado y comparado las ejecuciones.

Arquitectura:	x86_64
CPU(s):	12
Hilos por núcleo:	2
Procesador:	Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz
L1d cache:	192 KiB
L1i cache:	192 KiB
L2 cache:	1.5 MiB
L3 cache:	15 MiB

Escenario con costes de operaciones cambiado (problema2.prob):

OBS: (0,1);(0,3);(1,3)
SAT1: 1;4;1;2;8
SAT2: 6;1;1;3;8

En este caso hemos pensado en hacer que SAT1 le cueste mucho realizar una transmisión por cualquier tipo de error que tenga en su red de comunicaciones, y en cambio hemos hecho que SAT2 pueda transmitir fácilmente, pero le cueste mucho observar, lo que esperamos en este caso es que SAT1 deba hacer alguna operación de carga antes de observar por razones obvias, mientras que SAT2 las tendría que hacer antes de transmitir. Para evitar que el problema crezca exponencialmente, hemos aumentado la recarga de la batería junto con los costes de las operaciones, para que así no obtengamos un nº de nodos generados que no estarían dentro de un número factible.

Escenario con baterías cambiadas (problema3.prob):

OBS: (0,1);(0,3);(1,3)

SAT1: 1;1;1;1;1

SAT2: 1;1;1;1;1

En este caso se ha puesto a ambos satélites con escasa batería para que no sea posible realizar todas las observaciones sin necesidad de carga, de esta manera podemos observar que comportamiento tendrá el algoritmo y si los satélites aprovechan la batería o la malgastan. Las observaciones se mantienen en las mismas posiciones, pero en este caso la batería que tienen los satélites se ajusta a poder realizar una operación y necesariamente tengan que cargar o realizar IDLE.

Escenario añadir observaciones (problema4.prob):

OBS: (0,1);(0,3);(1,3);(2,8)

SAT1: 1;1;1;1;1

SAT2: 1;1;1;1;8

Partiendo del estado base se añade una nueva observación en la última hora observable del día, por lo que necesariamente deberán realizar 11 operaciones hasta esa observación, además de que deben pasar la noche para poder transmitir.

Escenario quitar observaciones (problema5.prob):

OBS: (0,1);(1,3)

SAT1: 1;1;1;1;1

SAT2: 1;1;1;1;8

En este caso se procederá a eliminar una observación para poder ver si en efecto el nº de nodos expandidos disminuye, teniendo en cuenta que las observaciones del problema base se encuentran en las posiciones iniciales. Se elimina una observación que sólo podía realizar el SAT 1, para que al menos SAT2 tenga la posibilidad de observar una.

Escenario con observación inobservable (problema6.prob):

OBS: (0,1);(0,3);(13,3)

SAT1: 1;1;1;1;1

SAT2: 1;1;1;1;8

Introducción de una observación en la posición 13, que constituye un ciclo de noche para los satélites, por lo que nos debe saltar una excepción indicando que no es válida una de las observaciones.

bruta:

Para el escenario dado en el enunciado (problema1.prob):

Tiempo Total: 44.4

Coste Total: 8

Longitud del plan: 8

Nodos expandidos: 8279

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: Gira
3. SAT1: Carga SAT2: Gira
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: Gira
6. SAT1: Transmite SAT2: Gira
7. SAT1: Carga SAT2: Gira
8. SAT1: Transmite SAT2: Transmite

Escenario con costes de operaciones cambiado (problema2.prob):

Tiempo Total: 8.9

Coste Total: 6

Longitud del plan: 6

Nodos expandidos: 4131

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Transmite SAT2: IDLE
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: IDLE
6. SAT1: Transmite SAT2: Transmite

Escenario con baterías cambiadas (problema3.prob):

Tiempo Total: 1.3

Coste Total: 8

Longitud del plan: 8

Nodos expandidos: 1938

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Carga SAT2: Carga
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: IDLE
6. SAT1: Transmite SAT2: IDLE
7. SAT1: Carga SAT2: Carga
8. SAT1: Transmite SAT2: Transmite

Escenario añadir observaciones (problema4.prob):

Tiempo Total: 704.4

Coste Total: 10

Longitud del plan: 10

Nodos expandidos: 25572

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: Gira
3. SAT1: Carga SAT2: Gira
4. SAT1: Observa SAT2: Observa
5. SAT1: IDLE SAT2: Gira
6. SAT1: IDLE SAT2: Gira
7. SAT1: Carga SAT2: Carga
8. SAT1: Transmite SAT2: Transmite
9. SAT1: Carga SAT2: Observa
10. SAT1: Transmite SAT2: Transmite

Escenario quitar observaciones (problema5.prob):

Tiempo Total: 0.2

Coste Total: 5

Longitud del plan: 5

Nodos expandidos: 714

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: Gira
3. SAT1: IDLE SAT2: Gira
4. SAT1: Carga SAT2: Observa
5. SAT1: Transmite SAT2: Transmite

Escenario con observación inobservable (problema6.prob):

Salta la excepción esperada.

lessOBS:

Para el escenario dado en el enunciado:

Tiempo Total: 1.6

Coste Total: 8

Longitud del plan: 8

Nodos expandidos: 1867

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: Gira
3. SAT1: Carga SAT2: Gira
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: Transmite
6. SAT1: Transmite SAT2: Gira
7. SAT1: Carga SAT2: Gira
8. SAT1: Transmite SAT2: Gira

Escenario con costes de operaciones cambiado (problema2.prob):

Tiempo Total: 0.0

Coste Total: 6

Longitud del plan: 6

Nodos expandidos: 158

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Transmite SAT2: IDLE
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: Transmite
6. SAT1: Transmite SAT2: IDLE

Escenario con baterías cambiadas (problema3.prob):

Tiempo Total: 0.1

Coste Total: 8

Longitud del plan: 8

Nodos expandidos: 509

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Carga SAT2: Carga
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: Carga
6. SAT1: Transmite SAT2: Transmite
7. SAT1: Carga SAT2: IDLE
8. SAT1: Transmite SAT2: IDLE

Escenario añadir observaciones (problema4.prob):

Tiempo Total: 16.8

Coste Total: 10

Longitud del plan: 10

Nodos expandidos: 5009

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: Gira
3. SAT1: Carga SAT2: Gira
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: Transmite
6. SAT1: Transmite SAT2: Gira
7. SAT1: Carga SAT2: Gira
8. SAT1: Transmite SAT2: Carga
9. SAT1: IDLE SAT2: Observa
10. SAT1: IDLE SAT2: Transmite

Escenario quitar observaciones (problema5.prob):

Tiempo Total: 0.0

Coste Total: 5

Longitud del plan: 5

Nodos expandidos: 240

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: Gira
3. SAT1: IDLE SAT2: Gira
4. SAT1: Carga SAT2: Observa
5. SAT1: Transmite SAT2: Transmite

Escenario con observación inobservable (problema6.prob):

Salta la excepción esperada.

distanciaUbeda:

Para el escenario dado en el enunciado:

Tiempo Total: 0.7

Coste Total: 8

Longitud del plan: 8

Nodos expandidos: 1214

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Carga SAT2: IDLE
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: Gira
6. SAT1: Transmite SAT2: Gira
7. SAT1: Carga SAT2: Gira
8. SAT1: Transmite SAT2: Transmite

Escenario con costes de operaciones cambiado (problema2.prob):

Tiempo Total: 0.1

Coste Total: 6

Longitud del plan: 6

Nodos expandidos: 418

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Transmite SAT2: IDLE
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: IDLE
6. SAT1: Transmite SAT2: Transmite

Escenario con baterías cambiadas (problema3.prob):

Tiempo Total: 0.1

Coste Total: 8

Longitud del plan: 8

Nodos expandidos: 389

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Carga SAT2: Carga
4. SAT1: Observa SAT2: Observa
5. SAT1: Carga SAT2: IDLE
6. SAT1: Transmite SAT2: IDLE
7. SAT1: Carga SAT2: Carga
8. SAT1: Transmite SAT2: Transmite

Escenario añadir observaciones (problema4.prob):

Tiempo Total: 27.6

Coste Total: 10

Longitud del plan: 10

Nodos expandidos: 5860

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: Carga SAT2: IDLE
4. SAT1: Observa SAT2: Observa
5. SAT1: IDLE SAT2: Gira
6. SAT1: Carga SAT2: Gira
7. SAT1: Transmite SAT2: Gira
8. SAT1: Carga SAT2: Transmite
9. SAT1: IDLE SAT2: Observa
10. SAT1: Transmite SAT2: Transmite

Escenario quitar observaciones (problema5.prob):

Tiempo Total: 0.0

Coste Total: 5

Longitud del plan: 5

Nodos expandidos: 211

1. SAT1: IDLE SAT2: Gira
2. SAT1: Observa SAT2: IDLE
3. SAT1: IDLE SAT2: IDLE
4. SAT1: Carga SAT2: Observa
5. SAT1: Transmite SAT2: Transmite

Escenario con observación inobservable(problema6.prob):

Salta la excepción esperada.

Conclusiones de las pruebas

En cuanto al segundo escenario, al cambiar los costes de las operaciones nos hemos dado cuenta de que el valor de recarga de batería es algo crucial, ya que si por ejemplo tenemos costes altos en las operaciones y solo podemos recargar muy pocas unidades, el número de nodos expandidos es exageradamente elevado comparado a aumentar el nivel de recargas de la batería, por tanto, si lo que deseamos es obtener un problema con un número más limitado de nodos, lo ideal sería aumentar la recarga en la operación cargar.

En lo referente al siguiente escenario, podemos ver que si ponemos una batería con pocas unidades de energía el nº de pasos (profundidad del algoritmo) necesarios para obtener nuestra solución va a aumentar, ya que se llamará a la operación carga muy a menudo, en este caso hemos intentado aplicar el caso más extremo, una única unidad de batería. Aquí cada vez que un SAT realiza una operación debe realizar carga, por lo que para un problema que necesitaría de 4 pasos con un nivel de batería más admisible, en nuestra prueba se necesitan 8.

En el cuarto escenario hemos añadido una observación más, como era de esperar el nº de pasos aumenta, y el nº de nodos expandidos lo hace también considerablemente, el resultado es el esperado. Esto nos hace pensar lo mucho que puede cambiar la rapidez del algoritmo simplemente añadiendo una variable más, ya que cada iteración extra que se añade genera un gran número de nuevos nodos que poder visitar, por tanto, la calidad de las heurísticas es vital para intentar reducir este factor lo máximo posible.

El quinto escenario es una extensión del anterior, simplemente hemos quitado una observación respecto al problema base, pero aquí el resultado podría haber sido muy diferente. En este caso, las observaciones están en las posiciones iniciales de nuestro 'grid', pero si hubiera una sola observación, en este caso en las posiciones finales del tablero, el nº de nodos expandidos sería parecido al del caso anterior ya que tendríamos que recorrer casi todas las posiciones para poder captar la única observación del tablero.

En lo referente al último escenario, hemos captado mediante nuestro código un error que maneje cualquier observación que se encuentre en una franja no observable y así evitar que se tenga que recorrer todos los nodos, lo que supondría una cantidad de tiempo bastante considerable.

Viendo los resultados de las pruebas realizadas podemos llegar a la conclusión de que normalmente lessOBS es una mejor heurística que distanciaUbeda ya que el nº de observaciones restantes es una variable que afecta en mayor medida al estado del problema que la distancia restante de un satélite a una observación. El problema está planteado de tal manera que da igual que satélite haga la observación, lo que realmente importa es que esa observación sea tomada y luego transmitida.

HERRAMIENTAS USADAS

Las herramientas usadas han sido principalmente Ubuntu y Atom. En cuanto a Ubuntu, nos ha permitido seguir conociendo las características que puede ofrecer este sistema operativo y sobre todo a usar su terminal de manera efectiva. Por otro lado, hemos usado Atom como procesador de texto y para que los dos integrantes del grupo puedan compartir los archivos y editarlos al mismo tiempo y de esta manera poder comentar ideas al momento. Por último, GitHub nos ha sido de gran utilidad ya que hemos podido subir diferentes versiones de nuestro proyecto y tenerlas accesibles en todo momento.

CONCLUSIONES

En definitiva, esta práctica nos ha ayudado a conocer la utilidad de las heurísticas y cómo pueden mejorar la resolución de los problemas. Esto también ha conllevado que aprendamos a cómo obtener una heurística admisible e informada. El proyecto ha sido codificado en python, lenguaje que no conocíamos muy bien, pero que ahora estamos algo más familiarizados con él. El trabajo por tanto nos ha ayudado a conocer más el entorno de las búsquedas heurísticas y a refinar el uso de otras herramientas y conocimientos que seguro que nos serán útiles en un futuro próximo.