Heurística y Optimización

Curso 2020/2021

Práctica 2: Satisfacción de restricciones y búsqueda heurística.

Desarrollado por:

Marcelino Tena Blanco; NIA: 100383266

Alejandro Díaz García; NIA: 100383181

Índice

Introducción

En la siguiente memoria se va a documentar el desarrollo de la práctica 2 de la asignatura Heurística y Optimización orientada a resolver dos problemas, uno de satisfacción de restricciones y otro de búsqueda heurística. Para la implementación de una solución del problema de CSP se va a codificar en python usando la librería python-constraint, la cual nos permite modelar en el lenguaje un problema de satisfacibilidad de restricciones introduciendo en él variables y restricciones correspondientes al problema, para así posteriormente obtener las posibles soluciones que satisfagan dichas restricciones. En cambio para el segundo problema se va a realizar una búsqueda informada utilizando el algoritmo visto en clase A*. Crearemos dos heurísticas diferentes que logren resolver el problema correctamente.

El problema de satisfacción trata de asignar a diferentes satélites en franjas horarias distintas una antena para poder realizar sus funciones, para cada satélite en cada franja tendrá visibles una serie de antenas con las que poder comunicarse. El fin es determinar qué configuraciones satisfacen las condiciones impuestas en el problema y saber que antena tiene asignada cada satélite en cada franja.

En cuanto al problema de búsqueda, se trata de determinar qué pasos deben dar los satélites en cada hora dentro de las franjas para lograr observar todos los objetos observables y poder transmitirlos de forma satisfactoria. Para ellos se intentará hacer en el menor tiempo posible, tomando las mejores decisiones en cada momento y teniendo como guía la función heurística que evaluará en cada estado lo lejos o cerca que se está del estado final.

Tras esta breve introducción se realizará una descripción sobre los modelos de cada uno de los problemas, así como detalles a destacar en su implementación. También se mostrarán los resultados obtenidos en la ejecución de las implementaciones así como un análisis. Por último concluimos con unas conclusiones sobre cada una de las partes de la práctica, así como unas conclusiones generales.

Descripción de los modelos

Parte 1: Asignación de antenas de transmisión a satélites

Introducción

En el problema 1 nos piden un modelo de satisfacción de restricciones. El problema trata sobre la comunicación entre satélites y antenas. Un satélite tiene asignado varias antenas, las cuales solo pueden mandar información en una franja de tiempo determinada:

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

Tabla de datos sobre la asignación de satélites respecto a la franja de tiempo y las antenas a las que se puede comunicar.

Por lo tanto, lo que debemos hacer es asignar satélites a antenas según las siguientes restricciones:

1. Todos los satélites deben tener asignados una antena, siempre que tengan una franja horaria visible.
2. SAT1 Y SAT2 deben tener asignadas la misma antena.
3. SAT2, SAT4 y SAT5 deben tener asignadas diferentes antenas.
4. Si a SAT5 se le asigna ANT12, entonces SAT4 no se le puede asignar ANT11.
5. Si las antenas ANT7 y ANT12 son asignadas, se le debe seleccionar satélites con franja horaria que comience antes de las 12 o después de las 12.

Modelización del problema

Estamos ante un problema de satisfacción de restricciones, donde tenemos un conjunto de variables (satélites), con un dominio para cada variable, donde el dominio para los satélites son las antenas. Asignar una antena para cada satélite

- Variables: Satélites

$$X = (SAT1,\ SAT2,\ SAT3.1,\ SAT3.2,\ SAT4,\ SAT5,\ SAT6.1,\ SAT6.2)$$

Establecemos variables como satélites, excepto porque debemos agregar dos satélites más debido a que SAT3 y SAT6 tiene dos dominios totalmente distintos, por lo que consideramos que existen dos variables más, cada uno con su propio dominio: SAT3.1, SAT3.2, SAT6.1 y SAT6.2. Esta decisión ha sido tomada principalmente debido a que trabajan en diferentes rangos horarios y de cara a la modelización simplifica mucho el problema al verlo como si fueran otros satélites más. Luego las soluciones tendrán la pertinente interpretación en las variables de los satélites .1 y .2

- Dominio: Antenas

$$D_{SAT1} = \{ANT1,\ ANT2,\ ANT3,\ ANT4\}$$
$$D_{SAT2} = \{ANT1,\ ANT2,\ ANT3\}$$
$$D_{SAT3.1} = \{ANT4,\ ANT6\}$$
$$D_{SAT3.2} = \{ANT7,\ ANT9,\ ANT10\}$$
$$D_{SAT4} = \{ANT8,\ ANT11,\ ANT12\}$$
$$D_{SAT5} = \{ANT1,\ ANT7,\ ANT12\}$$
$$D_{SAT6.1} = \{ANT7,\ ANT9\}$$
$$D_{SAT6.2} = \{ANT3,\ ANT4,\ ANT5\}$$

De esta forma, la **restricción 1** está resuelta de forma implícita porque cada variable sólo debe seleccionar una antena de su dominio por solución, por lo que quedaría modelada de dicha manera.

- Restricciones
  - **Restricción 2**: SAT1 y SAT2 deben tener asignadas la misma antena

$$R_{SAT1,SAT2} = \{(x,y)\}$$
$$Donde\ x = y,\ x \in D_{SAT1} \wedge y \in D_{SAT2}$$

  - **Restricción 3**: SAT2, SAT4 y SAT5 deben tener asignadas diferentes antenas

$$R_{SAT2,SAT4,SAT5} = \{(x,y,z)\}$$
$$Donde\ x \neq y \neq z,\ x \in D_{SAT2},\ y \in D_{SAT4},\ z \in D_{SAT5}$$

  - **Restricción 4**: Si a SAT5 se le asigna ANT12, entonces SAT4 no se le puede asignar ANT11.

$$R_{SAT5},\ R_{SAT4} = \{(ANT12,x),(y,ANT11)\}$$
$$Donde\ x \neq ANT11 \wedge y \neq ANT12,\ x \in D_{SAT4},\ y \in D_{SAT5}$$

  - **Restricción 5**: Si en una solución se asignan las antenas ANT 7 y ANT 12, ambas deben estar en la misma franja horaria. Existen dos franjas, la de antes de las 12:00 y la de después de las 12:00. Por lo tanto para las variables cuyas antenas están en sus dominios, podemos sacar los siguientes conjuntos dependiendo de la franja en la que operen dichos satélites.

    FranjaAntes = { SAT5, SAT6.1}
    FranjaDespués = {SAT3.2, SAT4}

    Por lo que podemos sacar las siguientes restricciones:

$$R_{SAT5,SAT3.2} = \{(ANT12,x), (y,ANT7)\}$$
$$Donde\ x \neq ANT7 \wedge y \neq ANT12,\ x \in D_{SAT3.2},\ y \in D_{SAT5}$$
$$R_{SAT5,SAT4} = \{(ANT12,x), (y,ANT7)\}$$

$$Donde\ x \neq ANT7 \wedge y \neq ANT12,\ x \in D_{SAT4},\ y \in D_{SAT5}$$

$$R_{SAT6.1,SAT4} = \{(ANT12,x), (y,ANT7)\}$$
$$Donde\ x \neq ANT7 \wedge y \neq ANT12,\ x \in D_{SAT4},\ y \in D_{SAT6.1}$$

Estas restricciones cumplirían con la especificación de no tener ANT7 y ANT12 asignadas a una solución de distinta franja.

Implementación del problema

Respecto a la implementación del problema utilizando la librería de python-constraint se ha optado por ciertas restricciones utilizar las funciones que esta librería nos proporciona en lugar de realizar una implementación de las mismas. Esto solo se ha dado en las restricciones 2 y 3, para las que se ha utilizado AllEqualConstraint() para la restricción 2 y AllDifferentConstraint() para la restricción 3. Como su nombre indica la función AllEqualConstraint() implica que se elijan elementos comunes del dominó para ambas variables (SAT1 y SAT2), por otra parte y de forma similar trabaja AllDifferentConstraint() otorgando valores de los dominios de las variables de tal modo que sean diferentes, por lo que no tendrán ninguna antena en común asignada SAT2, SAT4 y SAT5.
Para la implementación de la cuarta restricción se ha optado por descartar aquellas soluciones que cumplan con que si SAT5 se comunica con ANT12, entonces SAT4 se comunica con ANT11, con esto eliminamos dichas soluciones.
En cuanto a la última de las restricciones se ha decidido identificar aquellas combinaciones las cuales supondrán que en ANT7 y ANT12 estuvieran en franjas horarias completamente distintas y se ha decidido eliminarlas de las soluciones, tal y como se ha hecho en la restricción anterior. Para la salida de todas las soluciones se ha decidido hacerlo mediante un fichero de salida solutions.txt, para que se logre sacar correctamente se requiere tener instalada la librería *numpy*. Por pantalla solo se imprime una de las soluciones así como el número de soluciones que se obtienen.

Casos de prueba

Para realizar los casos de prueba hemos realizado la siguiente tabla donde:

Nombre del caso	Modo de probarlo	Esperamos	Resultado que obtenemos
Funcionamiento correcto	Vamos a ejecutar el código original de forma que funcione como se espera	Esperamos que se cumplan todas las restricciones.	468 soluciones Las soluciones cumplen las restricciones.
Error restricción 2 general	Cambiamos el dominio de SAT1 y SAT2 para que la intersección sea el conjunto vacío.	Esperamos que no se encuentre solución para CSP.	0 soluciones No cumple con la primera restricción, por lo que no se puede satisfacer.
Error restricción 3 general	Cambiamos el dominio de SAT2, SAT4, SAT5, haciendo que los tres sean el mismo.	Esperamos que no se encuentre solución para CSP.	0 soluciones No cumple con la tercera restricción, por lo que no se puede satisfacer.
Error restricción 4 general	SAT5 sólo tiene un elemento que es ANT12 y SAT4 sólo tiene un elemento que es ANT11.	Esperamos que no se encuentre solución para CSP.	0 soluciones No cumple con la cuarta restricción, por lo que no se puede satisfacer.
Restricción 4 SAT4 debe elegir ANT11	El dominio de SAT5 solo tiene un elemento: ANT12.	Esperamos que SAT4 no seleccione ANT11.	72 soluciones SAT4 no elige no elige ANT11 solo ANT8.
Restricción 4 probar que SAT5 no elija ANT12	El dominio de SAT4 solo tiene un elemento: ANT11.	Esperamos que SAT5 no seleccione ANT12.	180 soluciones SAT5 no elige en ANT12, solo ANT1 y ANT7.
Restricción 2 y restricción 3, mediante SAT2	SAT2 tiene 3 antenas. SAT4 y SAT5 tienen 2 antenas donde ambas son iguales y del dominio de SAT2. SAT1 tiene todas las antenas.	Esperamos que encuentre solución, la cual es que SAT1 y SAT2 tienen asignada la misma antena.	72 soluciones Efectivamente todas las soluciones salen con SAT1 y SAT2 con la misma antena asignada.
Error restricción 2 y restricción 3, mediante SAT2	SAT2 tiene 3 antenas. SAT1, SAT4, SAT5 tienen 2 antenas ambas son iguales y del dominio de SAT2.	Esperamos que no encuentre solución debido porque las antenas de SAT1 no las puede elegir SAT2 porque las tienen elegidas SAT4 y SAT5	0 soluciones No encuentra ninguna solución, por lo que no se puede satisfacer el problema.
Restricción 4 probar SAT5	El dominio de SAT5 no puede contener a ANT12.	Esperamos que funcione correctamente	386 soluciones Se cumplen todas las restricciones.
Restricción 4 probar SAT4	El dominio de SAT4 no puede contener a ANT11.	Esperamos que funcione correctamente	288 soluciones Se cumplen todas las restricciones.

Parte 2: Búsqueda Heurística

Introducción

El problema propuesto para la segunda parte de la práctica se trata de observar unos elementos en un terreno, los cuales llamaremos observables, y transmitir esas observaciones. Para ello, usaremos dos satélites que podrán reconocer dos filas de terreno a la vez. El algoritmo deberá asignar una operación para cada satélite en un tiempo determinado:

- **Observar**: analiza un elemento del terreno, el cual se notifica a los satélites mediante unas coordenadas en un fichero. Solo puede observar un elemento en una hora.
- **Transmitir**: envía la información analizada a la base de datos. Tiene memoria suficiente para guardar todos los observables en caso de no retransmitir nunca. Solo puede transmitir un elemento en una hora.
- **Girar**: el satélite se desplaza hacia una banda superior o inferior para poder observar otros elementos.
- **Recargar**: el satélite obtiene energía para poder realizar las anteriores operaciones.
- **Ocioso/Idle**: el satélite no hace nada.

Cada una de las operaciones, excepto recargar e idle, cuestan energía. Los gastos de energía según la operación y la capacidad máxima se pasan a través de un fichero.

Por lo tanto, debemos realizar heurísticas capaces de reconocer cuál operación es más importante que otra y reconocer de antemano si se puede realizar dicha operación.

El problema se considera resuelto si todos los observables están leídos y enviados a la base de datos.

Hemos definido que el coste del problema son las horas totales que han pasado desde el principio. Hay que recordar que los satélites solo trabajan doce horas al día, por lo que cuando se sobrepasa las 12 horas, hay que añadir 12 horas más al coste.

Modelización del problema

Para este problema vamos a utilizar el algoritmo A*, el cual es un algoritmo de búsqueda completo, es decir, si existe una solución siempre la encontrará. El algoritmo A* es especial debido a que utiliza la táctica del *primero mejor.* Por una parte la búsqueda la realiza en el menor tiempo posible, pero por la otra el camino otorgado no es siempre el mejor.

Para realizar la búsqueda heurística hemos definido un conjunto de estados, incluyendo un estado inicial y final, un conjunto de cinco posibles acciones por satélite y dos heurísticas diferentes, donde una resulta más compleja que la otra.

En nuestro caso hemos decidido guardar en cada estado las variables que cambian, como puede ser el terreno, las operaciones que realiza cada satélite, la batería de cada satélite o la lista de retransmisiones de cada uno.

- **Estado**:

$$Estado = (Hora,\ MObservables\ ,\ SAT_1,\ SAT_2\ )$$
$$Hora = x,\ donde\ x \in \{0, 1, 2, ...11\}$$
$$MObservables = (\ Banda_0,\ Banda_1,\ Banda_2,\ Banda_3)$$
$$Banda = (a_1,\ a_2,\ ...,\ a_i)\ donde\ a_i \in \{0, O_k\},\ \forall i \in \{0, ..., 11\},\ \forall k \in \{0, ..., 47\}$$
$$SAT = (Bater\acute{i}aDisponible,\ BandasActuales,\ Retransmisiones,\ Operación)$$

$$BandasActuales = (x, y) \ donde \ \forall x \in \{0, 1, 2\}, \ \forall y \in \{x + 1\}$$
$$Retransmisiones = (O_1, O_2, ..., O_j) \ \forall j \in \{0, ..., 47\}$$
$$Operación = x, \ donde \ \forall x \in \{Giro, IDLE, Observar, Transmitir, Recargar\}$$

- **Hora**: La hora en la que está el estado.
- **MObservables**: Lista de bandas en las que está dividido el terreno.
- **Banda**: Lista de elementos llamados O que contiene un terreno.
- **a**: Indica mediante un uno si hay que observar la celda o con O_j si existe algún elemento que observar.
- **O_j**: Es el nombre dado del elemento que debemos observar, donde j es el número del elemento.
- **SAT**: Es un satélite que tiene unas propiedades.
- **BateríaDisponible**: La cantidad actual de energía que tiene para su uso.
- **BandasActuales**: Bandas en las que está trabajando el satélite en cierto estado.
- **Retransmisiones**: Las retransmisiones que tiene pendientes de realizar tras haberlas observado.
- **Operación**: La operación que ha realizado en el estado.

El **estado inicial** tiene: Hora es 0; MObservables es la matriz más en las celdas que nos pasan por fichero añadir que tienen observables; Bandas orígenes, que son para Sat1 [0,1] y para Sat2 [2,3]; la batería de los dos satélites la obtengo mediante la batería total por el fichero; la operación de los dos satélites al iniciar el IDLE; y las retransmisiones están vacías.
El **estado final** se llega cuando la matriz de observables no contiene ningún observable y las listas de retransmisiones no contiene ninguna transmisión

- **Variables Globales independientes al estado**
  - $(GastoObservable_i, \ GastoRetransmitir_i, \ GastoGirar_i, \ udsbateria_i, \ BandasOriginales_i)$
    $donde \ \forall i \in \{0, \ 1\}$
- **Acciones**
  - **Observar**: el satélite obtiene información sobre un observable y lo elimina de la tabla de observables en el estado actual. Se debe tener en cuenta que para poder observar debe de haber un observable en una de las bandas y en la hora donde está el satélite y debe tener batería para realizar la acción.
    - **Operador** : Observar(Sat, MObservables, Hora)
    - **Precondiciones**:

$$BateríaDisponible_i \geq GastoObservable_i \wedge MObservables[banda_{ij}][Hora] \neq 0$$

    - **Efectos:**

$$Retransmisiones_{ik} = MObservables\,[banda_{ij}][Hora] \wedge MObservables\,[banda_{ij}][Hora] = 0$$
$$donde \ \forall i \in \{0, 1\}, \forall j \in \{0, 1\} \wedge k = |Retransmisiones_i| + 1$$

  - **Retransmitir:** el satélite envía información a la estación base y elimina esa información de las retransmisiones. Se debe tener en cuenta que existan elementos en la lista de retransmisiones y que tenga suficiente batería para realizar la acción.
    - **Operador** : Retransmitir(Sat)
    - **Precondiciones**:

$$BateríaDisponible_i \geq GastoRetransmitir_i \wedge |Retransmisiones_i| > 0$$

- **Efectos:**

$$Retransmisiones_i = Retransmisiones_i - Retransmisiones_{i0}$$
$$donde \ \forall i \in \{0, 1\}$$

- **Girar:** el satélite se cambia a una banda superior o inferior. Se debe tener en cuenta que solo se puede hacer una vez, que existe límites para girar, que si gira después de haber girado una primera vez, debe volver a sus bandas iniciales y que si tiene suficiente batería para poder realizar la acción. Esta acción se divide en tres operadores:
  - **Operador** : Girar(Sat, bandasOriginales)
  - **Precondiciones**:

    BA = bandasActuales$_{i;}$; BO = bandasOriginales$_i$ donde $\forall i \in \{0, 1\}$

$$BateríaDisponible_i \geq GastoGirar_i \wedge \forall i \in \{0, 1\}$$
$$(BO = BA \wedge (\ (\ min(BO) \leq min(BA)\ ) \vee (\ max(BO) \leq max(BA)\ )\ )\ ) \vee BO \neq BA$$

  - **Efectos:**

$$(BO = BA \wedge (\ min(BO) < min(BA)\ ) \Rightarrow BA_0 = BA_0 - 1 \wedge BA_1 = BA_1 - 1$$
$$(BO = BA \wedge (\ max(BO) > max(BA)\ ) \Rightarrow BA_0 = BA_0 + 1 \wedge BA_1 = BA_1 + 1$$
$$(BO \neq BA \Rightarrow BA_0 = BO_0 \wedge BA_1 = BO_1$$

- **Recargar**: obtiene unas unidades de energía específicas para recarga la batería del satélite. Se tiene que tener en cuenta que el satélite debe tener menos de la batería total.
  - **Operador** : Recargar(Sat)
  - **Precondiciones**:

$$BateríaDisponible_i < TotalBateria_i$$

  - **Efectos:**

$$BateriaDisponible_i = BateríaDisponible + udsbateria$$
$$BateriaDisponible_i > TotalBateria_i \Rightarrow BateriaDisponible_i = TotalBateria_i$$
$$donde \ \forall i \in \{0, 1\}$$

- **Idle**: el satélite no hace nada y deja que pase la hora. No se tiene que tener en cuenta nada.
  - **Operador** : idle(Sat)
  - **Precondiciones** y **Efectos:** Ninguno

Para cada acción realizada se aumenta una unidad el coste, dado el avance del tiempo.
Las heurísticas que han sido definidas son las siguientes:

- **Heurística 1**:

La heurística 1 se basa en el conocimiento que se tiene del terreno y la cantidad de observaciones que deben ser realizadas por los satélites y en las observaciones que tienen los satélites pendientes.

$$h1(n) = NObservables * 10 + |Retransmisiones_k| * 5 + |Retransmisiones_k| * 5 \ \forall k \in \{0, 1\}$$

$$donde\ NObservables = \sum_{i=0}^{|MObservables|} \sum_{j=0}^{|Banda_i|} x,\ donde\ x = 1\ si\ MObservables_{ij} \neq 0 \wedge x = 0\ si\ MObservables_{ij} = 0$$

La heurística tendrá un valor más alto mientras haya algún observable pendiente de ser observado y menor mientras estén en la lista por retransmitir, hasta tener un valor de 0 que significa que se han observado y transmitido todos los observables del terreno.

- **Heurística 2**:

La heurística 2 se basa en la distancia (tiempo) a la que se encuentra cada satélite de los puntos observables. Por lo tanto, realizamos un sumatorio de las distancias que tienen los satélites a los observables, más las distancias de bandas que tienen de la banda en la que se encuentran los satélites a la banda en la que se encuentran los observables, es decir se tiene en cuenta que están en la banda correcta para llevar a cabo la observación. En líneas generales esta heurística irá decreciendo en función de lo cerca que se encuentren los satélites para realizar las observaciones pendientes, hasta tener una distancia de 0 dado que no queda ningún elemento por observar.

$$h2(n) = SumaDistancias + |Retransmisiones_0| + |Retransmisiones_1| + DiferenciaBandas$$

$$SumaDistancias = \sum_{i=0}^{NObservables} Distancia(Hora,\ Observable)$$

$$Observable = x\ /\ \forall x \in MObservables_{ij},\ si\ MObservables_{ij} \neq 0$$

$$donde\ NObservables = \sum_{i=0}^{|MObservables|} \sum_{j=0}^{|Banda_i|} x,\ donde\ x = 1\ si\ MObservables_{ij} \neq 0 \wedge x = 0\ si\ MObservables_{ij} = 0$$

$$DiferenciaBandas = \sum_{i=0}^{|bObs|} min(a(bObs_ibs - bandasActuales_0),\ abs(abs(bObs_i - bandasActuales_1))$$

$$bObs_k = i,\ \ si\ MObservables_{ij} \neq 0\ \ \forall k \in \{0, ..., NObservables\}, \forall i \in \{0, ..., |Bandas|\} \wedge \forall k \in \{0, ..., 11\}$$

Se hace un sumatorio de las distancias de los satélites para que cada vez que se esté más cerca de realizar una observación la haga con la finalidad de disminuir el cómputo total de distancias a objetos, dado que en el momento en el que no haya distancias habrán sido observados todos los puntos del terreno. Por otro lado, para tener en cuenta las transmisiones también se penaliza a los estados en los que no se realicen transmisiones pudiéndose hacer.

Implementación del problema

La implementación del algoritmo A* junto con el resto de aspectos a tener en cuenta del problema de búsqueda se ha decidido realizar en el lenguaje de programación Python, dada la simplicidad sintáctica del mismo y las facilidades para manejar cierto tipo de datos.
En nuestro caso, un nodo es un elemento del árbol que se compone de un estado, de función heurística, de valor heurístico, de coste y de un puntero hacia el nodo padre.
La implementación tiene 3 ficheros:

- **main.py**: es el fichero donde se ejecuta el programa principal. Lee el fichero de entrada, genera el nodo inicial y llama a aStar para realizar el algoritmo y obtener los datos. Una vez acabado el algoritmo imprime las estadísticas y la solución
- **aStar.py**: realiza todas las acciones que son necesarias para buscar la solución:
    - **Algoritmo**: realiza el algoritmo A*. Simplificadamente, mientras que no haya solución o la lista abierta se quede sin nodos, entonces sacará un nodo de la lista

abierta y comprobará si ese nodo es igual a uno ya abierto (obtenido de la lista cerrada). Si ya hay uno igual y con menor coste, entonces no hará nada más. Si no existe ninguno igual, entonces mirará si es final. Si no es final, expandirá el árbol creando nuevos nodos. Si es final, el último nodo obtenido será el nodo final.

- **Crear nodos**: genera nodos a partir de un nodo padre. Estos nodos son conjuntos de operaciones de los satélites con una nueva matriz de observables. En el caso de que se puedan realizar todas las operaciones para todos los satélites, el máximo de nodos creados serán veinticinco (cinco operaciones elevadas a dos satélites). Este caso no se suele dar. Al final de crear los nodos, se ordenan según la función heurística (de menor a mayor) y se agregan en la lista abierta.

- **nodo.py:** La clase nodo aporta la información necesaria de los estados con los que va a trabajar el algoritmo A*, en esta clase se tiene una referencia al nodo padre del mismo, al tratarse de una estructura en forma de árbol. Tenemos la información de los satélites, las heurísticas, la función de coste y la función de evaluación.
  - **Coste (g)**: Es el total de horas pasadas. Los satélites están disponibles solo 12 horas al día, entonces cuando pasen esas horas hay que aumentar el coste en 12 horas más.
  - **evaluarh1**: La función de la heurística 1 va obteniendo el cardinal de observables que quedan por ser observados y lo multiplica por 10 para tener mas penalización, luego hace lo mismo con las listas de observables que están por retransmitir dentro de los satélites, multiplicándolos por 5, esto tiene menos penalización respecto a que no se hayan observado aún.
  - **evaluarh2**: La función de la heurística 2 va sumando la distancia de tiempo entre donde están los satélites y donde están los observables. Para ello verifica si está aún por pasar por el observable para simplemente restar, en caso contrario, de que el satélite haya pasado de largo el observable, se calcula la distancia completa hasta llegar de nuevo al observable en las siguientes horas. Para la diferencia de bandas se ha usado la distancia mínima y en valor absoluto. Por último para garantizar que el algoritmo acabe transmitiendo se le ha dado una penalización menor si quedan observaciones por transmitir al igual que en la heurística h1.
  - **satelite.py**: objeto satélite. Tiene todas las variables que necesita un satélite que son la lista de retransmisiones pendiente; la operación que realiza, que en caso de que realice observar o transmitir se guardará en una nueva variable el objeto; la batería que le queda; y las bandas actuales. Tiene otras variables, pero estas son utilizadas para organizar.

Para poder ejecutar el programa es necesario:

- python 3.8.5 y Tener instalada la librería Numpy
- ./Comos.sh <fichero de entrada> <heurística>
  - <fichero de entrada>: variables iniciales para el nodo inicial
  - <heurística>: 0: no usar ninguna heurística; 1: usar la heurística 1; 2: usar la heurística 2

Análisis de los resultados

Parte 1: Asignación de antenas de transmisión a satélites

Tras realizar la ejecución de la implementación realizada se han obtenido un total de 468 soluciones, para comprobar si dichas soluciones son correctas se van a coger de forma aleatoria soluciones y se van a comprobar una a una si cumplen con las restricciones descritas. Por otra parte, en caso de no existir ningún tipo de restricciones en el problema el número de soluciones sería de 3888 soluciones en caso de tener asignada una antena para cada satélite, por lo que se considera que el resultado obtenido es satisfactorio tras acotar considerablemente el número de soluciones.

Tras elegir 5 soluciones de forma aleatoria del fichero de salida, del total de 468 podemos verificar que satisfacen las restricciones del problema.

Solución 01:

{'SAT4': 'ANT12', 'SAT5': 'ANT1', 'SAT2': 'ANT3', 'SAT1': 'ANT3', 'SAT6.1': 'ANT9', 'SAT3.2': 'ANT10', 'SAT3.1': 'ANT6', 'SAT6.2': 'ANT5'}

R1: Si ; R2: Si ; R3: Si ; R4: Si; R5: Si

Solución 02:

{'SAT4': 'ANT11', 'SAT5': 'ANT7', 'SAT2': 'ANT3', 'SAT1': 'ANT3', 'SAT6.1': 'ANT9', 'SAT3.2': 'ANT7', 'SAT3.1': 'ANT6', 'SAT6.2': 'ANT3'}

R1: Si ; R2: Si ; R3: Si ; R4: Si; R5: Si

Solución 03:

{'SAT4': 'ANT11', 'SAT5': 'ANT1', 'SAT2': 'ANT2', 'SAT1': 'ANT2', 'SAT6.1': 'ANT7', 'SAT3.2': 'ANT10', 'SAT3.1': 'ANT4', 'SAT6.2': 'ANT4'}

R1: Si ; R2: Si ; R3: Si ; R4: Si; R5: Si

Solución 04:

{'SAT4': 'ANT8', 'SAT5': 'ANT7', 'SAT2': 'ANT3', 'SAT1': 'ANT3', 'SAT6.1': 'ANT9', 'SAT3.2': 'ANT9', 'SAT3.1': 'ANT6', 'SAT6.2': 'ANT4'}

Solución 05:

{'SAT4': 'ANT8', 'SAT5': 'ANT1', 'SAT2': 'ANT2', 'SAT1': 'ANT2', 'SAT6.1': 'ANT7', 'SAT3.2': 'ANT7', 'SAT3.1': 'ANT4', 'SAT6.2': 'ANT3'}

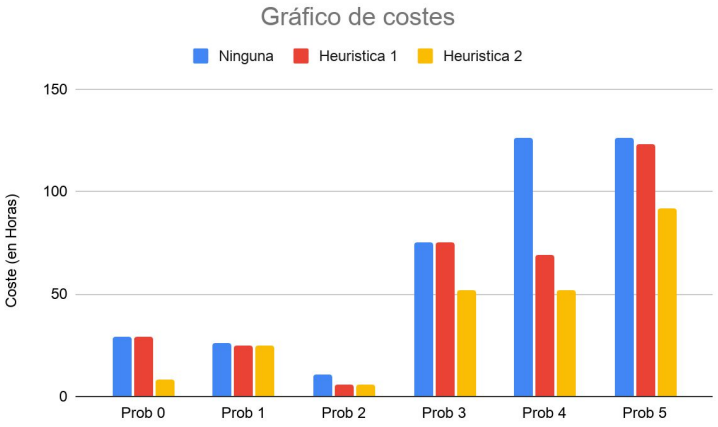
R1: Si ; R2: Si ; R3: Si ; R4: Si; R5: Si

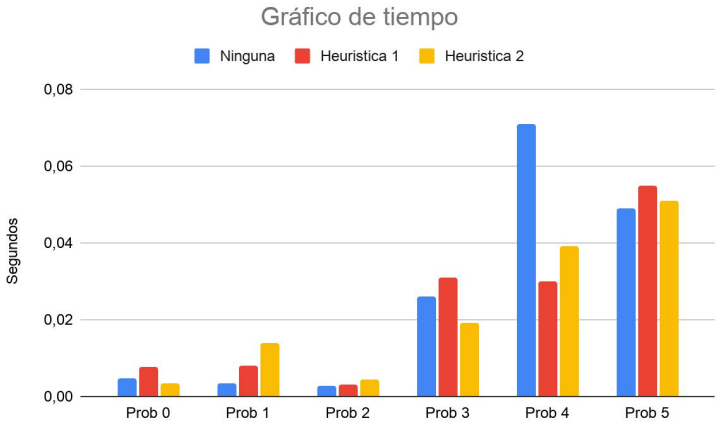
Parte 2: Búsqueda Heurística

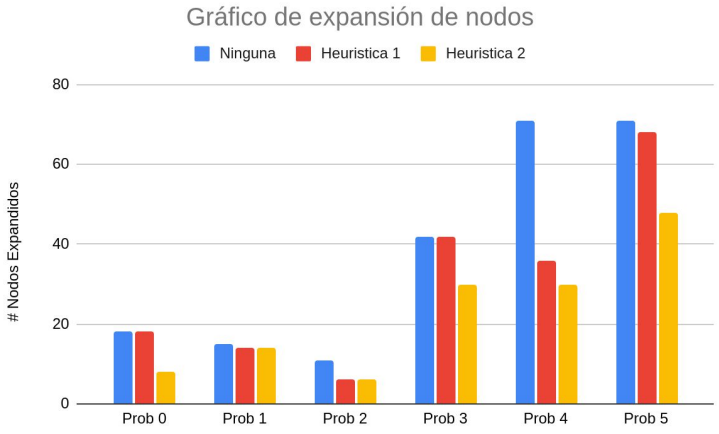
Se van a comparar los tres casos con los que se pueden obtener resultados con la implementación realizada, ejecución sin función heurística, con función heurística 1 y con la función heurística 2. Posteriormente se van a obtener los resultados de las ejecuciones.

Han sido creados 6 ficheros cada uno con una configuración distinta de problema acorde al formato indicado en el enunciado. Lo que se ha buscado con estos ficheros ha sido el ver la optimización que se tiene al usar las heurísticas y medirlas según su rendimiento. Los ficheros son los siguientes: problema0.prob, que se trata de la configuración propuesta en el enunciado; problema1.prob, añadiendo más observaciones y aumentando la batería de los satélites a 99; problema2.prob, configuración similar a problema0 pero con baterías al 99; problema3.prob, configuración de observadores similar a problema1 pero con solo 1 unidad de capacidad de batería cada satélite.

Entrando en problemas más costosos tenemos problema4.prob el cual cuenta con 24 puntos de observación repartidos y 100 de capacidad cada satélite, problema5.prob con la misma cantidad de puntos de observación que problema4 pero con 10 de capacidad de batería cada satélite. Dichos archivos son adjuntados dentro de la carpeta de la práctica.







Como se puede apreciar en las gráficas presentadas la heurística 2 mejora considerablemente el rendimiento en la mayoría de los problemas planteados encontrando una solución en un menor coste. En cuanto al tiempo también se nota una mejoría en algunos casos con la heurística 2, pero en otros se ve perjudicada. Respecto a la heurística 1 nos presenta buenos resultados en los casos más extremos en los que se tiene gran número de elementos por observar. A nivel global se puede apreciar claramente que la mejor opción para ejecutar los problemas en la mayoría de los casos es utilizando la heurística 2, siempre y cuando queramos obtener el menor coste. La heurística 2 nos proporciona unos resultados bastante satisfactorios guiando correctamente al algoritmo. De forma similar pasa con el gráfico de nodos expandidos lo que ocurre con el de costes, esto es debido al diseño planteado para la función de costes la cual se basa en las horas y cada nodo tiene como peculiaridad que su profundidad corresponde también con la hora, excepto en las franjas en las que no son operables los satélites.

Conclusiones

Conclusiones de la parte 1: CSP

Respecto a la realización de la parte 1 ha sido de gran ayuda el utilizar la librería de python-constraint dado que de no haberlo hecho en cualquier otro caso, habría sido más complejo de implementar. Por lo tanto ha sido de gran utilidad haber podido conocer el funcionamiento de la misma. En el apartado teórico nos ha ayudado sobre todo a comprender la cantidad de variantes que pueden llegar a satisfacer un problema real con tal cantidad de restricciones.

Conclusiones de la parte 2: Búsqueda heurística

En la parte 2 hemos podido reforzar los conocimientos sobre búsquedas ya dados en otras asignaturas, pero ha sido en esta donde hemos podido aprender más en detalle la importancia de las heurísticas en las búsquedas informadas, tanto a nivel de implementación como teórico. El haber podido implementar el algoritmo A Star ha reforzado y complementado correctamente los conocimientos obtenidos sobre algoritmos de búsqueda vistos en las clases teóricas.
Para mantener la coherencia de la práctica entera, tras realizar la parte 1 en el lenguaje de programación Python, también hemos adoptado este para la realización de esta segunda parte, por lo que nuestra habilidad con el lenguaje también se ha visto incrementada. La parte que más se debe destacar es la importancia de realizar un desarrollo incremental de la parte, dada la complejidad y la cantidad de subproblemas que han ido surgiendo.

Conclusiones generales y problemas encontrados

En esta segunda práctica de la asignatura Heurística y Optimización hemos podido poner en práctica ciertos conceptos de los bloques de satisfacción de restricciones y de búsqueda informada. En general ha sido una práctica que nos ha resultado muy interesante de realizar y sobre la que hemos aprendido bastantes conceptos mientras se realizaba. A nivel de la asignatura nos ha sido de gran utilidad e interés el haber podido aprender todo lo dado en las prácticas.