Heurística y Optimización

Universidad Carlos III de Madrid

Práctica 2

Satisfacción de Restricciones y Búsqueda Heurística

Autores:

Jorge Villaseñor Catalán NIA: 100383398
100383398@alumnos.uc3m.es

Manuel Ramos González NIA: 100383545
100383545@alumnos.uc3m.es

Índice

1. Introducción

En el presente documento se testifica el proceso de desarrollo de la segunda práctica. Esta introducción sirve a modo de explicación de los contenidos del documento para clarificar la estructura del mismo.

En el segundo punto de esta memoria se describen los modelos que hemos utilizado para representar los problemas de la parte uno y de la parte dos. En el tercer punto se muestran los casos de prueba llevados a cabo y su posterior análisis, tanto de la parte uno como de la parte dos. Finalmente se muestran las conclusiones acerca de esta práctica.

2. Descripción de los modelos

2.1. Modelo de la parte 1

Variables y dominios:

Para modelar el problema hemos separado este en dos partes:

- Asignaturas que imparte cada profesor:

  Por cada profesor hemos definido dos variables, esto lo hemos realizado así debido a que cada profesor tiene que ocuparse de dos asignaturas.
  Las variables que representan a Juan son: J1 y J2.
  Las variables que representan a Lucía son: L1 y L2.
  Las variables que representan a Andrea son: A1 y A2.

  $X = \{J1, J2, L1, L2, A1, A2\}$

  El dominio de todas estas variables está formado por todas las asignaturas disponibles: matemáticas (M), ciencias de la naturaleza (CN), ciencias sociales (CS), lengua y literatura (LL), inglés (I) y educación física (EF).

  $D_i = \{M, CN, CS, LL, I, EF\} \quad \forall i \in X$

- Calendario de asignaturas:

  Por cada asignatura hemos definido dos variables excepto para educación física que solo hemos definido una. Esto lo hemos realizado así debido a que

cada asignatura tiene dos horas semanales excepto educación física que solo tiene una.
Las variables que representan matemáticas son: M1 y M2.
Las variables que representan ciencias de la naturaleza son: CN1 y CN2.
Las variables que representan ciencias sociales son: CS1 y CS2.
Las variables que representan lengua y literatura son: LL1 y LL2.
Las variables que representan inglés son: I1 e I2.
La variable que representa educación física es: EF.

y = {M1, M2, CS1, CS2, CN1, CN2, LL1, LL2, I1, I2, EF}

El dominio de todas las variables excepto las de matemáticas y ciencias sociales está formado por las horas de clase disponibles: Lun1, Lun2, Lun3, Mar1, Mar2, Mar3, Mier1, Mier2, Mier3, Jue1 y Jue2.

D_i = {Lun1, Lun2, Lun3, Mar1, Mar2, Mar3, Mier1, Mier2, Mier3, Jue1, Jue2}

i $\in$ {CN1, CN2, LL1, LL2, I1, I2, EF}

El dominio de las variables de matemáticas es distinto ya que se indica en el enunciado que estas clases sólo pueden darse las primera horas, por lo que el dominio será: Lun1, Mar1, Mier1 y Jue1.

D_i = {Lun1, Mar1, Mier1, Jue1}

i $\in$ {M1, M2}

Para ciencias sociales ocurre lo mismo solo que esta solo puede darse las últimas horas, por lo que su dominio será: Lun3, Mar3, Mier3 y Jue2.

D_i = {Lun3, Mar3, Mier3, Jue1, Jue2}

i $\in$ {CS1, CS2}

Restricciones:

La restricción *"La duración de cada una de las clases es de 1 hora de duración, en la que sólo se puede impartir una única materia."* la hemos implementado simplemente obligando a que cada variable de asignatura adquiera un valor diferente a las otras.

La restricción "*Para todas las materias se deben impartir 2 horas semanales, excepto para Educación Física que solo tiene asignada 1 hora semanal*" la hemos implementado implícitamente a la hora de definir las variables de las asignaturas ya que cada asignatura tiene dos variables excepto educación física que tiene solo una.

La restricción "*las 2 horas dedicadas a Ciencias de la Naturaleza se deben impartir de forma consecutiva el mismo día*" la hemos implementado obligando a que si alguna de las variables de ciencias de la naturaleza tiene un valor, el valor de la otra variable sea obligatoriamente el de una hora inmediatamente posterior o anterior al valor de la primera.

La restricción "*La materia de Matemáticas no puede impartirse el mismo día que Ciencias de la Naturaleza e Inglés*" la hemos implementado obligando a que si el valor de las variables de matemáticas son unos días específicos, el valor de las variables de inglés y ciencias de la naturaleza no puedan adquirir ningún valor que se corresponda al mismo día que el valor de las variables de matemáticas.

La restricción "*además, la materia de Matemáticas debe impartirse en las primeras horas, y la de Ciencias Sociales en las últimas*" la hemos implementado restringiendo el dominio de las variables de matemáticas y ciencias sociales.

La restricción "*cada profesor debe impartir 2 materias, que son diferentes a las de sus compañeros*" la hemos implementado simplemente obligando a que cada variable de profesor adquiera un valor diferente al resto.

La restricción "*Lucía solo se encargara de Ciencias Sociales, si Andrea se encarga de Educación Física*" la hemos implementado obligando a que alguna de las variables de Lucía pueda tomar el valor de ciencias sociales solo si alguna de las variables de Andrea tiene el valor de educación física.

La restricción "*Juan no quiere encargarse de Ciencias de la Naturaleza o de Ciencias Sociales, si algunas de sus horas se imparte a primera hora los lunes y jueves*" se ha implementado obligando a que si el valor de alguna de las variables de ciencias naturales adquiere un valor referido a la primera hora de cualquier día, ninguna de las variables de Juan pueda adquirir el valor de ciencias naturales (CN). La asignatura de ciencias sociales no se ha tenido en cuenta ya que en el dominio de sus variables no hay ningún valor que sea a primera hora.

2.2. Modelo de la parte 2

Modelando el problema del cálculo de rutas propuesto en esta práctica como un problema de búsqueda heurística nos queda lo siguiente:

Estados:

([PosBus, AOcupados], [Niños], NiñosPorEntregar)

donde, [PosBus, AOcupados] es una lista con dos elementos. El primero es un string que indica la posición en la que está el bus en ese estado (ej.: “P1”). El segundo es una variable entera que indica el número de niños que están dentro del bus.

[Niños] es una lista que contiene la información de los niños. Cada posición de esta lista es a su vez otro lista con dos elementos que muestra las característica del niño en cuestión
La primera posición de esta lista es un string que indica si el niño está esperando en una parada, si está en dentro del bus o si está entregado. En el caso de que esté esperando en una parada, se muestra que parada es. La segunda posición de la lista que define a un niño en concreto es un string que indica a que colegia pertenece dicho niño. Unos ejemplos de una posición en concreto de la lista [Niños] serían los siguientes:
[“P2”, “C3”] este niño está esperando la parada 2 y va al colegio 3
[“Entregado”, “C1”] este niño ya ha sido entregado y pertenece al colegio 1
[“Bus”, “C2”] este niño está dentro del bus y pertenece al colegio 2
Así pues, un ejemplo de la lista [Niños] sería el siguiente:
[[“P2”, “C3”], [“Entregado”, “C1”], [“Bus”, “C2”]]

NiñosPorEntregar es un entero que indica el número de niños que quedan por ser entregados en sus respectivos colegios. Hemos decidido incluir esta información en la representación de los estados para no tener que recorrer la lista [Niños] cada vez que se quiera comprobar cuántos niños faltan por entregar. Cuando un niño es entregado, se decrementa el valor de esta variable.

Habiendo explicado como hemos representado los estados, un ejemplo de estado inicial y final serían los siguientes:
Estado Inicial: ([“P1”, 0], [[“P2”, “C3”], [“P1”, “C1”], [“P3”, “C2”]], 3)
Estado Final: ([“P1”, 0], [[“Entregado”, “C3”], [“Entregado”, “C1”], [“Entregado”, “C2”]], 0)

Acciones:

Hemos definido tres acciones que se pueden ejecutar, mover el autobus, recoger a los niños de una parada y dejar a los niños en sus respectivos colegios. En la siguiente tabla se muestran dichas acciones, las precondiciones que han de darse para poder aplicar las acciones y los efectos o cambios que se producen cuando se ejecutan dichas acciones sobre un estado.

Acción	Precondiciones	Efectos
Mover_Autobús	La única precondición para poder realizar esta acción es que desde la parada actual del bus se pueda llegar a una o más paradas. Por cada parada a la que se pueda mover se crea un sucesor	El valor PosBus del estado obtendrá el valor de la parada a la que se mueve el autobús.
Recoger_Niños	Las precondiciones son las siguientes: - El autobús tiene que estar en una parada en la que haya al menos un niño esperando - El número de niños que lleve el autobús no puede ser el máximo	El autobús recogerá a todos los niños que quepan en el autobús. - Si esta cantidad es n, el valor de la situación de los n primeros niños de la lista [Niños] del estado actual que estén en la parada actual del autobús cambiará a “Bus”. -El valor de AOcupados aumenta en función del número de niños que se recogen.
Dejar_Niños	Las precondiciones son las siguientes: - El autobús tiene que estar en una parada en la que haya un colegio - El autobús debe llevar al menos un niño perteneciente a dicho colegio	El autobús dejará a todos los niños que pueda. - El valor de la situación de los niños que estén en el autobús y cuyo colegio esté en la posición actual del bus cambiará a “Entregado”. - El valor de AOcupados disminuye en función del número de niños que se dejan. - El valor de NiñosPorEntregar disminuye en función del número de niños que se dejan.

Hemos asumido que el coste de ejecutar las acciones es el siguiente:
El coste de Mover_Autobús desde una parada a otra es igual al peso de la arista que los une en el grafo que representa el problema.

El coste de Recoger_Niños depende del número de niños que se recojan. Así pues, recoger un niño tiene coste igual a uno, recoger dos niños tiene coste igual a dos, y así sucesivamente.

El coste de Dejar_Niños depende del número de niños que se vayan a dejar. Así pues, dejar un niño tiene coste uno, dejar dos niños tiene coste dos, y así sucesivamente.

Hemos decidido asignar estos costes a las acciones de Recoger_Niños y Dejar_Niños para aproximarnos más a lo que ocurriría en la vida real.
En la vida real, recoger a 1 niño tiene un coste menor que recoger a 20 niños. Seguramente los 20 niños tarden más tiempo en subirse al autobús.
De esta manera asumimos que todos los niños tardan el mismo tiempo en subirse al autobús y que el coste de dichas acciones crece linealmente en función del número de niños a recoger/dejar.

No obstante, a pesar de que recoger solo un niño cuesta menos que recoger a 20 niños que estén esperando en una parada, el hecho de que en las paradas siguientes haya un colegio donde poder dejar a todos esos niños hace que sea más eficiente coger a todos los niños que entren en el bus. Es por eso que en nuestro programa hacemos que el bus recoja tantos niños como pueda en una parada en función de sus plazas libres.

Heurísticas:

Para resolver este problema haciendo uso del Algoritmo A* hay que definir una heurística que ha de ser admisible si queremos encontrar la solución óptima.
Para encontrar algunas heurísticas hemos hecho uso de la técnica de relajación de restricciones, eliminando algunas de las precondiciones de las acciones para llegar a un problema óptimamente relajado y poder estimar cuál es el coste desde un estado hasta la meta.

<u>Heurística 1</u>:

- Esta heurística se calcula con el número de niños que quedan por entregar, es decir, los niños que están en una parada más los niños que están dentro del bus. La heurística queda como sigue:
  h(n) = NiñosPorEntregar.
  Tal y como hemos definido los costes de las acciones Recoger_Niños y Dejar_Niños, en el que por cada niño que esté esperando en una parada su coste de subirlo y bajarlo del bus es 2, haciendo el recuento de los niños que

aún no han sido entregados en sus colegios nos aseguramos que nunca sobrestimamos el coste real de llegar a la meta, consiguiendo así una heurística admisible.

- Para aplicar esta heurística, el nombre del parámetro <heuristica> que se tiene que pasar a la hora de realizar la ejecución es “heuristica1”.

Heurística 2:

- Esta heurística se calcula como la suma de los niños que quedan por entregar y los niños que están esperando en una parada. Para ello recorremos la lista de los niños y en un contador vamos guardando cuantos están en una parada y luego se lo sumamos a NiñosPorEntregar.
  Como NiñosPorEntregar = niños esperando en paradas + niños en el bus
  la heurística nos queda de la siguiente forma:
  h(n) = 2 * número niños esperando en paradas + niños que están en el bus
  Como el coste de recoger a un niño más el coste de dejarlo es 2, decidimos contar el número de niños que están esperando en una parada dos veces. Esto nos garantiza acercarnos más al coste real de un estado a la solución pero sin sobrestimarlo haciendo que esta heurística sea de nuevo admisible.

- Para aplicar esta heurística, el nombre del parámetro <heuristica> que se tiene que pasar a la hora de realizar la ejecución es “heuristica2 “.

Si no se quiere utilizar ninguna de las heurísticas anteriormente mencionadas, simplemente con poner en el parámetro <heuristica> cualquier cosa que no sea ni heuristica1 ni heuristica2 se asigna h(n) = 0 por defecto. De esta manera, se ejecutará el algoritmo de Dijkstra.
Cabe destacar que la implementación del Algoritmo A* que hemos realizado pertenece al segundo pseudocódigo explicado en la clase de ejercicios, es decir, la implementación menos eficiente en memoria pero que ejecuta en menos tiempo.

3. Análisis de los resultados

3.1. Análisis de la parte 1

Para el problema original, el número de soluciones posibles obtenido es 8999424. Creemos que es una cantidad tan grande debido a como hemos modelado el problema. Debido a que por cada profesor definimos dos variables y por cada asignatura lo mismo si en una solución la variable J1 adquiere el valor 'LL' y J2 adquiere el valor 'M' y en otra solución J1 adquiere el valor 'M' y J2 el valor 'LL', técnicamente sería la misma solución pero en nuestro problema cuenta como 2 y eso pasa con todas las variables excepto la de educación física. A continuación hemos puesto dos soluciones diferentes del problema y como hemos podido comprobar cumplen todas las restricciones:

Solución 1:

Horas	Lunes	Martes	Miércoles	Jueves
09:00-10:00	EF	M2	M1	CN2
10:00-11:00	I2	LL2	LL1	CN1
11:00-12:00	I1	CS2	CS1	

Juan: Se encargará de Lengua y literatura y Matemáticas
Lucía: Se encargará de Ciencias sociales y Ciencias naturales
Andrea: Se encargará de Educación física e Inglés

Solución 2:

Horas	Lunes	Martes	Miércoles	Jueves
09:00-10:00	M1	CN2	M2	I1
10:00-11:00	EF	CN1	LL2	I2
11:00-12:00	CS2	LL1	CS1	

Juan: Se encargará de Lengua y literatura y Ciencias sociales
Lucía: Se encargará de Matemáticas y Educación física
Andrea: Se encargará de Ciencias naturales e Inglés

A continuación vamos a mostrar diferentes casos de prueba que hemos realizado para ver qué supone la adición y eliminación de algunas restricciones y pondremos algunas soluciones para ver si efectivamente se cumple la nueva restricción en el caso de que añadamos alguna:

Para comenzar hemos añadido la restricción de que la asignatura de Educación física tenga que darse solo en las segundas horas. Para este caso el número de soluciones posibles ha disminuido a 5405184. Este hecho tiene sentido ya que al restringir más el problema el número de soluciones disminuye. Una posible solución añadiendo dicha restricción sería la siguiente:

Horas	Lunes	Martes	Miércoles	Jueves
09:00-10:00	I1	M2	I2	M1
10:00-11:00	CN1	LL1	LL2	EF
11:00-12:00	CN2	CS2	CS1	

Juan: Se encargará de Matemáticas y Educación Física
Lucía: Se encargará de Lengua y literatura y Ciencias Naturales
Andrea: Se encargará de Ciencias Sociales e Inglés

Otro de los caso de prueba que hemos realizado para verificar el funcionamiento de las restricciones mencionadas en el enunciado de esta práctica ha sido el siguiente: Incluimos un restricción más al problema original que impone que la asignatura de CN se imparta solo a las últimas horas de los días que hay clase.
Con este caso de prueba queremos comprobar que no encuentra solución ya que una de las restricciones del problema original es que las 2 horas de Ciencias naturales se impartan de manera consecutiva. Comprobamos, efectivamente, que no se encuentra solución al problema planteado.

Por último, probamos otro caso de prueba que resulta de la eliminación de la restricción "Lucía solo se encargará de Ciencias Sociales, si Andrea se encarga de Educación Física".
Esperamos que el tamaño del número de soluciones se vea incrementado dado que restringimos menos el problema. Tras ejecutar dicho problema obtenemos 14999040 soluciones diferentes

3.2. Análisis de la parte 2

Para los casos de prueba de esta sección hemos decidido hacer uso de 3 configuraciones diferentes del problema. Para cada prueba proporcionamos una captura del problema a resolver para poder visualizar mejor la situación del problema, una tabla para comparar los cambios producidos dependiendo de la heurística que se use y finalmente la ruta óptima que sigue el bus para alcanzar el estado final .

Prueba 1: problema1.prob

En esta primera prueba el problema a resolver es el presentado en el enunciado. La situación del problema es la que se muestra en la imagen:

```
    P1  P2  P3  P4  P5  P6  P7  P8  P9
P1  --  12  --  --  --  --  --  --   9
P2  12  --  13  --  10  --  --  --  --
P3  --  13  --  13  --  --  --  --   6
P4  --  --  13  --   8  --  --   9  --
P5  --  10  --   8  --  14  --   5  --
P6  --  --  --  --  14  --   7  --  --
P7  --  --  --  --  --   7  --   6  --
P8  --  --  --   9   5  --   6  --  --
P9   9  --   6  --  --  --  --  --  --
C1: P6; C2: P3; C3: P8
P2: 1 C2, 1 C3; P4: 1 C3; P5: 1 C2; P7: 2 C1, 1 C2
B: P1 5
```

	Sin heurística	Heurística 1	Heurística 2
Tiempo de ejecución	3.89110088348 s	3.53691196442 s	3.44649600983 s
Coste Total	116	116	116
Paradas visitadas	12	12	12
Nodos expandidos	2145	2006	1949

Solución:

P1 -> P2 (S: 1 C2, 1 C3) -> P5 -> P4 (S: 1 C3) -> P8 (B: 2 C3) -> P7 (S: 2 C1, 1 C2) -> P6 (B: 2 C1) -> P5 (S: 1 C2) -> P4 -> P3 (B: 3 C2) -> P9 -> P1

Prueba 2: problema2.prob

Para este problema hemos querido aumentar la capacidad máxima del autobús, también hemos aumentado considerablemente la cantidad de niños que hay en cada parada y se ha disminuido el número de paradas con respecto a la prueba anterior

```
    P1  P2  P3  P4  P5  P6
P1  --   8  --  --  20   5
P2   8  --  10  --  --  --
P3  --  10  --   3  --  --
P4  --  --   3  --  15  --
P5  20  --  --   15 --   3
P6   5  --  --  --   3  --
C1: P3; C2: P4
P2: 10 C1, 10 C2; P5: 10 C1, 10 C2; P6: 10 C1, 10 C2
B: P1 15
```

	Sin heurística	Heurística 1	Heurística 2
Tiempo de ejecución	111.646406889 s	100.998716116 s	92.7817208767 s
Coste Total	264	264	264
Paradas visitadas	19	19	19
Nodos expandidos	7977	7495	7060

Solución:

P1 -> P6 (S: 10 C1, 5 C2) -> P5 -> P4 (B: 5 C2) -> P3 (B: 10 C1) -> P2 (S: 10 C1, 5 C2) -> P3 (B: 10 C1) -> P4 (B: 5 C2) -> P5 (S: 10 C1, 5 C2) -> P4 (B: 5 C2) -> P3 (B: 10 C1) -> P2 (S: 5 C2) -> P1 -> P6 (S: 5 C2) -> P5 (S: 5 C2) -> P4 (B: 15 C2) -> P3 -> P2 -> P1

Prueba 3: problema3.prob

Para esta prueba hemos disminuido aún más el número de paradas y tambien hemos reducido el número de niños.

```
    P1  P2  P3  P4  P5
P1  --   8  --  --  20
P2   8  --  10  --  --
P3  --  10  --   3  --
P4  --  --   3  --  15
P5  20  --  --  15  --
C1: P3; C2: P4
P2: 5 C1, 1 C2; P4: 10 C1; P5: 2 C1, 2 C2
B: P1 5
```

	Sin heurística	Heurística 1	Heurística 2
Tiempo de ejecución	12.2804939747 s	11.3442180157 s	10.5041880608 s
Coste Total	144	144	144
Paradas visitadas	15	15	15
Nodos expandidos	3430	3226	3034

Solución:

P1 -> P2 (S: 5 C1) -> P3 (B: 5 C1) -> P4 (S: 5 C1) -> P3 (B: 5 C1) -> P4 (S: 5 C1) -> P3 (B: 5 C1) -> P2 (S: 1 C2) -> P3 -> P4 (B: 1 C2) -> P5 (S: 2 C1, 2 C2) -> P4 (B: 2 C2) -> P3 (B: 2 C1) -> P2 -> P1

Como se puede observar en los 3 casos de prueba ambas heurísticas son admisibles ya que no sobreestiman el coste real desde un nodo cualquiera hasta el nodo final y reducen la cantidad de nodos expandidos. La heurística 2 está mejor informada que la 1 ya que se acerca más al coste real desde un nodo cualquiera hasta la meta pero siempre sin sobreestimarlo y además facilita que el Algoritmo de A* expanda menos nodos. El coste total de la solución y el número de paradas visitadas en cada problema son el mismo independientemente de la heurística, esto es debido a que la solución siempre es la óptima debido a que las heurísticas son admisibles.

4. Conclusiones

La realización de esta práctica nos ha permitido familiarizarnos más con los problemas de Satisfacción de Restricciones y los problemas de Búsqueda Heurística. En particular nos ha sido útil para conocer la implementación del Algoritmo A*. Nos ha resultado de ayuda trabajar con los nodos que entran y salen en la lista abierta y en la lista cerrada que utiliza este algoritmo.

Nos ha parecido muy curioso a la par que interesante el hecho de que los problemas propuestos en esta práctica puedan ser perfectamente problemas de la vida real, donde haya que planificar un horario de clase teniendo en cuenta las preferencias y limitaciones de cada uno de los profesores y de la preferencia a la que se quieran impartir las clases de las diferentes asignaturas. Así como el hecho de calcular la mejor ruta que un autobús debe de seguir para entregar a todos los alumnos que esperan ser recogidos y entregados en sus respectivos colegios.