MSEL
Modelling in Science Education and Learning

Modelling in Science Education and Learning
Volume 4, No. 21, 2011.
Instituto Universitario de Matemática Pura y Aplicada

Un problema a resolver con los algoritmos de caminos más cortos

Cristina Jordán, Jordi Burriel, Raquel Herráiz
Universidad Politécnica de Valencia
cjordan@mat.upv.es, jordi.bv@hotmail.com, raquel rhc@hotmail.com

Abstract

Numerosos problemas pertenecientes a los más diversos campos pueden ser resueltos a partir de la modelización en teoría de grafos, materia en pleno auge, podríamos decir que con crecimiento exponencial, por su amplia aplicabilidad. Uno de los primeros problemas que se plantean al alumno que inicia el estudio de la teoría de grafos es el cálculo del trazado de un camino de mínimo peso de un vértice a otro, es decir, el cálculo de la distancia mínima que separa dos vértices dados, así como el recorrido a realizar para obtenerla. Para ilustrarlo, motivar su estudio e iniciar al estudiante en el mundo de la modelización y su amplia gama de posibilidades, presentamos el siguiente caso concreto, en el que ayudamos a la policía a atrapar a los autores de un robo. La solución consiste en representar la ciudad en la que tiene lugar el atraco mediante un grafo no dirigido ponderado positivo, y aplicar el algoritmo de Floyd, entrelazado con razonamientos de tipo combinatorio. Podremos asegurar al final del ejercicio que los ladrones no tienen escapatoria, relacionando la solución con otro concepto de la teoría de grafos, la cortadura de vértices.

Graph theory solves, via a good modelization, a very large number of problems in different areas. This is why, this theory has been having an exponential increase in the last years. One of the basic problems that this theory solves is to obtain the shortest path between two points. In order to illustrate this problem, to motivate its study and to introduce the student in the world of the mathematical modelization and its large range of possibilities, we present a specific case: to help the police to catch the authors of a theft. The solution consists of representing the city where it took place by a no directed positive weighted graph and to apply the Floyd's algorithm mixed with a reasoning of combinatorial type. At the end of the exercise, we can assure that the thieves cannot escape, by using another concept of graph theory, the vertex-cut.

Keywords: Grafo ponderado, Algoritmo de Floyd, Problema camino más cortos.

1 Introducción

Actualmente la teoría de grafos es de utilidad en numerosos y variados campos ([2, 4]). Uno de los principales problemas a resolver es el conocido como el problema de los caminos más cortos, que como su nombre indica estudia cuál es y cuánto pesa el camino más corto entre dos vértices dados de un grafo ponderado, dirigido o no. Existen distintos algoritmos, con complejidad polinomial, que proporcionan una solución ([1, 3]).

No comporta grandes dificultades, por lo que se suele incorporar al temario de un primer curso sobre teoría de grafos. En nuestro caso, el ejemplo surgió del trabajo realizado en la asignatura Estructuras Matemáticas para la Informática 2, obligatoria de quinto cuatrimestre (tercer curso) del título de Ingeniero Informático, que se imparte en la ETSI Informática de la Universidad Politécnica de Valencia; asignatura de 4.5 créditos, distribuidos en 3 de teoría y 1.5 de laboratorio. En esta asignatura los alumnos tienen algunos conocimientos de grafos, pero es la primera vez que se enfrentan con problemas de modelización. El ejercicio que presentamos no es una aplicación directa del citado algoritmo, pudiéndose mostrar a un grupo de alumnos, ya sea de primer curso de carrera o superior, como ejemplo de que herramientas sencillas, conocidas de todos, combinadas de la forma adecuada, pueden ayudar a resolver un problema no trivial. Si el concepto de cortadura de vértices no es conocido por los estudiantes, puede suprimirse su alusión en la parte final del problema.

Aunque, como decíamos antes, la aplicación de la teoría de grafos se ha extendido rápidamente en los últimos años, siendo ampliamente utilizada en diferentes ramas de la ciencia, la introducción de su estudio como materia obligatoria en los planes de estudio de las diferentes ingenierías no ha ido parejo a esta expansión. Por ello en la primera sección introducimos la teoría básica necesaria para una adecuada comprensión de la modelización, constituyendo la sección 2 el enunciado del problema a resolver y la tres la solución de éste. El software que utilizamos para la aplicación del mencionado algoritmo es Mathematica 6©, siendo posible obtener los resultados que proporcionamos con la mayoría de software que permita la introducción de grafos.

2 Conceptos básicos de la teoría de grafos

Se llama grafo $G = (V, E)$, (ver [**?**]) a toda estructura formada por un conjunto de puntos no vacío, V, llamados vértices o nodos, y un conjunto E de pares no ordenados de puntos de V, llamados aristas. Se suele representar un grafo mediante un diagrama de puntos y líneas en el que los primeros representan a los vértices y una línea entre los puntos v_i y v_j representa la arista (v_i, v_j).

Un grafo $H = (V(H), E(H))$ es un subgrafo de $G = (V(G), E(G))$ si $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$.

Se llama camino a toda sucesión finita alterna de vértices y aristas, $v_1e_1v_2e_2 \dots v_{n-1}e_{n-1}v_ne_n$, donde $e_i = (v_i, v_{i+1})$, $i = 1, 2, ..., n-1$, en la que no se repite ningún vértice.

Un grafo es conexo si para cada par de vértices v_i y v_j existe un camino de v_i a v_j. Se llama componente conexa de G a todo subgrafo conexo de G que sea maximal respecto al conjunto de aristas.

En el caso de que cada arista tenga un valor asociado, al que llamamos peso de la arista, decimos que se trata de un grafo ponderado.

Un conocido problema de la teoría de grafos es el conocido como problema de los caminos más cortos. Una de sus versiones, a la que hacemos referencia en la siguiente sección, consiste en, partiendo de un grafo ponderado, determinar cuál es el camino más corto entre dos vértices dados y cuál es su peso. Existen varios algoritmos que resuelven el problema. En el presente artículo utilizamos el algoritmo de Floyd ([3]) que proporciona dos matrices, una de pesos y otra de vértices. El elemento p_{ij} de la matriz de pesos nos da el peso del camino más corto entre el vértice v_i y v_j, mientras que el elemento a_{ij} de la matriz de vértices nos indica quién es el anterior al vértice v_j en el camino más corto que lleva de v_i a v_j. Recurriendo reiteradamente a los elementos adecuados de la fila i de la matriz de vértices podemos, yendo hacia atrás, reconstruir el camino.

Para aplicar el algoritmo utilizamos el paquete de cálculo simbólico MATHEMATICA© 6.0.

Finalmente, un concepto que usamos hacia el final del problema es el de cortadura de vértices ([2, 3]). Se dice que un subconjunto V' del conjunto V es una cortadura de vértices cuando el grafo resultante de eliminar de G dichos vértices (y las aristas incidentes en ellos) sea no conexo.

3 Un problema de caminos más cortos

En el barrio ArcoIris, donde cada cruce recibe el nombre de un color, Paula y Pedro, tras una larga noche de fiesta y con unas copas de más, a primera hora de la mañana de vuelta a su casa, situada en la esquina Azul, deciden atracar una caseta de la ONCE que se encuentra en la esquina Lila, llevándose gran cantidad de números.

Mientras cuentan el número de billetes sustraídos, el vendedor llama a la comisaría de policía situada en la esquina Roja de este barrio. Tras los momentos de euforia, aunque no muy lúcidos todavía como consecuencia del alcohol ingerido, Paula y Pedro caen en la cuenta de que deberían retirarse del lugar lo antes posible; y deciden ir a ocultarse a su casa.

Dado que todo el barrio se encuentra en obras, la policía no dispone de vehículos que puedan desplazarse por la zona, por lo que en ese sentido se encuentra en las mismas condiciones que la pareja. Teniendo en cuenta las distancias entre cruces señaladas en el plano, ¿podrías ayudar a la policía a detener a los ladrones?

4 Solución al problema propuesto

En primer lugar, modelizamos el barrio mediante un grafo no dirigido ponderado, $G = (V, E)$, definido como,

$$\begin{aligned} V &= \{\text{esquinas de las calles}\} \\ &= \{\text{Marrón, Verde, Azul, Roja, Negra, Rosa, Lila, Amarilla, Naranja}\} \\ E &= \{(x, y) : x \text{ e } y \text{ son esquinas que se conectan a través de una calle}\} \\ p(x, y) &= \text{ distancia expresada en metros desde la esquina } x \text{ a la } y \end{aligned}$$

Consideramos los vértices numerados de la siguiente forma:

Marrón=1, Verde=2, Azul=3, Roja=4, Negra=5, Rosa=6, Amarilla=7, Lila=8, Naranja=9.

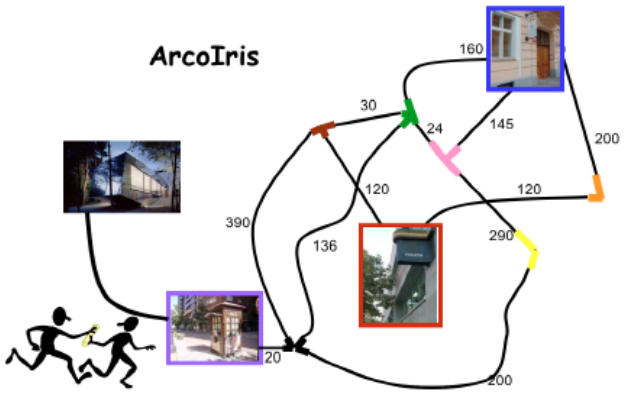

Figura 1: Plano del barrio ArcoIris

Para obtener una representación gráfica del anterior grafo, aplicamos los siguientes comandos de Mathematica©,

```
In[1]:= << Combinatorica`

In[74]:= G1 = SetEdgeWeights[Graph[{{{1, 2}}, {{1, 4}}, {{1, 5}}, {{2, 3}}, {{2, 5}}, {{2, 6}},
          {{3, 6}}, {{3, 9}}, {{4, 9}}, {{5, 7}}, {{5, 8}}, {{6, 7}}}, {{{0, 1}}, {{0.8, 1}},
          {{2.5, 1.7}}, {{0.7, -0.3}}, {{-0.5, -2}}, {{1, 0.8}}, {{2, -1}}, {{-1, -1.5}},
          {{3, -0.3}}}], {30, 120, 390, 160, 136, 24, 145, 200, 120, 200, 20, 290}];
     ShowGraph[SetEdgeLabels[G1, GetEdgeWeights[G1]],
      VertexLabel → {"", "", "Casa", "Policía", "", "", "", "Caseta ONCE", ""},
      VertexStyle → Disk[Large], EdgeLabel → Thickness[16], EdgeStyle → Thickness[0.01],
      VertexLabel → Thickness[Large],
      VertexColor → {(Blend[{Yellow, Purple}]), Green, Blue, Red, Black, Pink, Yellow,
        Purple, Orange}]
```

que proporcionan la representación gráfica

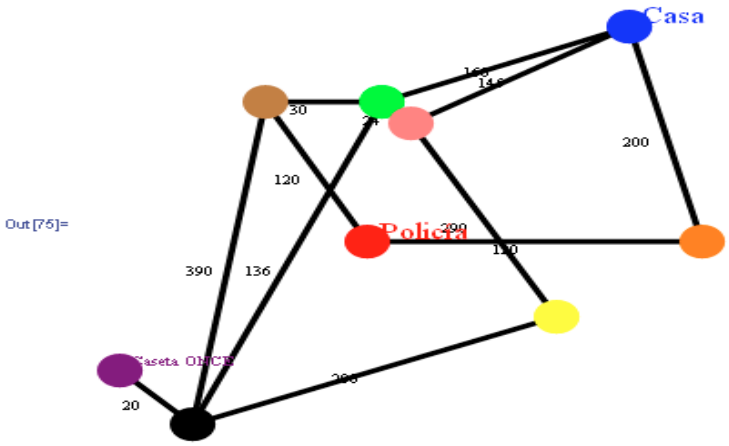

Figura 2: Modelización del barrio ArcoIris

Para obtener la información sobre los distintos recorridos que pueden ser de utilidad para los ladrones bastará con aplicar al grafo G el algoritmo de Floyd, que proporciona las siguientes matrices de pesos y vértices respectivamente. Recordemos que el elemento p_{ij} de la matriz de pesos (matriz de la izquierda) proporciona el peso del camino más corto para ir del vértice i al j. En cuanto a la matriz de la derecha o matriz de vértices, el elemento $a_{ij} = k$ indica que k es el vértice anterior al vértice j en el camino más corto del i al j. Buscando el elemento a_{ik}, $a_{ik} = h$, se obtiene que h es el vértice anterior al vértice k en el camino más corto del i al k. Continuando de forma análoga, conseguiremos obtener los vértices del camino más corto de i a j, es decir, $i, \ldots, h, k, j$.

```
In[20]:= prd1 = GraphDistanceMatrix[G1, Method → "FloydWarshall"] // MatrixForm
         prd2 = GraphDistanceMatrix[G1, Parent][[2]] // MatrixForm
```

Out[20]//MatrixForm=

0.	30.	190.	120.	166.	54.	344.	186.	240.
30.	0.	160.	150.	136.	24.	314.	156.	270.
190.	160.	0.	310.	296.	145.	435.	316.	200.
120.	150.	310.	0.	286.	174.	464.	306.	120.
166.	136.	296.	286.	0.	160.	200.	20.	406.
54.	24.	145.	174.	160.	0.	290.	180.	294.
344.	314.	435.	464.	200.	290.	0.	220.	584.
186.	156.	316.	306.	20.	180.	220.	0.	426.
240.	270.	200.	120.	406.	294.	584.	426.	0.

Out[21]//MatrixForm=

1	1	2	1	2	2	6	5	4
2	2	2	1	2	2	6	5	4
2	3	3	1	2	3	6	5	3
4	1	2	4	2	2	6	5	4
2	5	2	1	5	2	5	5	4
2	6	6	1	2	6	6	5	4
2	6	6	1	7	7	7	5	4
2	5	2	1	8	2	5	8	4
4	1	9	9	2	2	6	5	9

De la matriz de la derecha, aplicando lo comentado anteriormente, se obtiene que el vértice anterior al Azul o vértice 3, en el camino más corto del Lila o vértice 8, al Azul, es el Verde, es decir, el vértice 2, ya que la posición a_{83} de la mencionada matriz es 2. Repitiendo el proceso, la posición a_{82} indica que el vértice anterior al 2 en el camino más corto del 8 al 2 es el 5 (o

sea el Negro). Y, finalmente, como $a_{85} = 8$, el vértice anterior al 5 en el camino buscado es el propio 8. Por tanto, recopilando la información, el recorrido más corto para ir de la esquina Lila a la Azul es $8 - 5 - 2 - 3$, o lo que es lo mismo **Lila-Negra-Verde-Azul**.

Si buscamos en la matriz de la izquierda los pesos de las aristas mencionadas, (Lila,Negra), (Lila,Verde), (Lila,Azul), (respectivamente, elementos p_{85}, p_{82} y p_{83}, de la matriz) así como los pesos de los caminos más cortos desde la esquina Roja, donde se encuentra la comisaría, a las esquinas Negra, Verde y Azul (respectivamente, elementos p_{45}, p_{42} y p_{43}, de la matriz) obtenemos la siguiente tabla

	Roja	Lila
Negra	286	20
Verde	150!!!	156

Tabla 1. Coste de los caminos más cortos en el grafo de la Figura 2

donde finalmente no hemos reflejado el coste de ir de Lila o Roja a Azul, puesto que la policía puede apostar a un agente en la esquina Verde antes de que lleguen los ladrones, con lo que estos serían apresados. Como utilizando los caminos más cortos desde las esquina Roja y Lila a la Verde la policía atraparía a los ladrones, éstos deberían evitar pasar por dicha esquina. Así pues, la eliminamos del grafo que modeliza nuestro problema y obtenemos vía el programa Mathematica© el siguiente nuevo grafo,

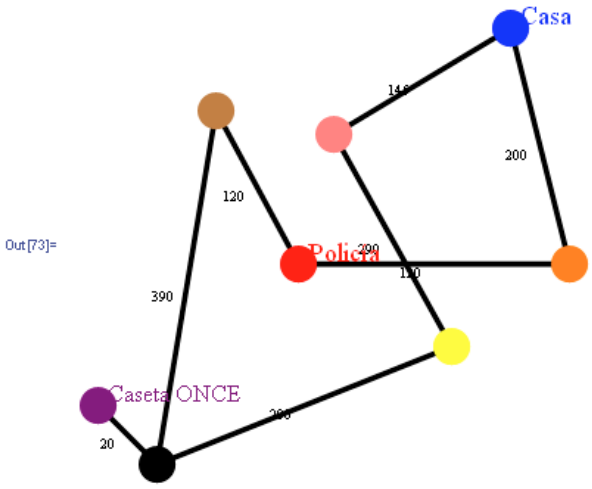

Figura 3: Grafo de la Figura 2 tras eliminar el vértice Verde

Volvemos a aplicar Floyd .

```
In[24]:= prd1 = GraphDistanceMatrix[G2, Method → "FloydWarshall"] // MatrixForm
         prd2 = GraphDistanceMatrix[G2, Parent][[2]] // MatrixForm
```

Out[24]//MatrixForm=

0.	440.	120.	390.	585.	590.	410.	240.
440.	0.	320.	635.	145.	435.	655.	200.
120.	320.	0.	510.	465.	710.	530.	120.
390.	635.	510.	0.	490.	200.	20.	630.
585.	145.	465.	490.	0.	290.	510.	345.
590.	435.	710.	200.	290.	0.	220.	635.
410.	655.	530.	20.	510.	220.	0.	650.
240.	200.	120.	630.	345.	635.	650.	0.

Out[25]//MatrixForm=

1	8	1	1	2	4	4	3
3	2	8	6	2	5	4	2
3	8	3	1	2	4	4	3
4	5	1	4	6	4	4	3
3	5	8	6	5	5	4	2
4	5	1	6	6	6	4	2
4	5	1	7	6	4	7	3
3	8	8	1	2	5	4	8

Y, de forma análoga al caso anterior, obtenemos que el camino más corto en este caso es

Lila-Negra-Amarilla-Rosa-Azul.

Reflejando en una tabla los pesos de los caminos más cortos desde las esquinas Lila y Roja a las esquinas intermedias, obtenemos la tabla

	Roja	Lila
Negra	510	20
Amarilla	710	220
Rosa	465!!!	510

TABLA 2. Coste de los caminos más cortos tras eliminar la esquina Verde

Al igual que en el caso anterior observamos que la policía llega antes que los atracadores a una de las esquinas intermedias, la Rosa en este caso. Por tanto, la esquina Rosa no les es de utilidad y rehacemos el plano sin incluirla.

El nuevo grafo a utilizar es

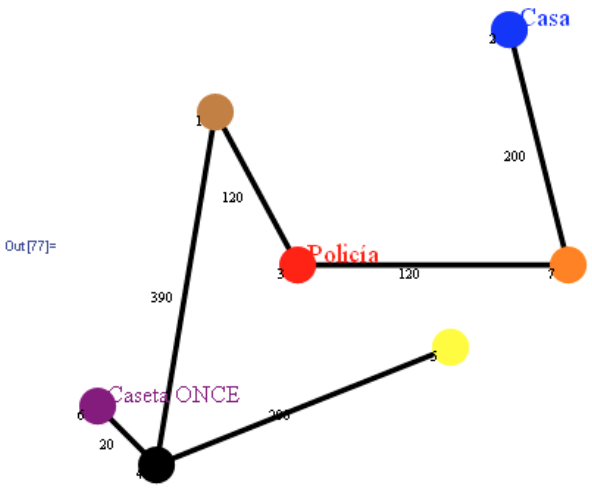

Figura 4: Grafo de la Figura 3 tras eliminar el vértice Rosa

Y la aplicación de Floyd

```
In[28]:= prd1 = GraphDistanceMatrix[G3, Method → "FloydWarshall"] // MatrixForm
         prd2 = GraphDistanceMatrix[G3, Parent][[2]] // MatrixForm
```

proporciona las matrices

Out[28]//MatrixForm=

0.	440.	120.	390.	590.	410.	240.
440.	0.	320.	830.	1030.	850.	200.
120.	320.	0.	510.	710.	530.	120.
390.	830.	510.	0.	200.	20.	630.
590.	1030.	710.	200.	0.	220.	830.
410.	850.	530.	20.	220.	0.	650.
240.	200.	120.	630.	830.	650.	0.

Out[29]//MatrixForm=

1	7	1	1	4	4	3
3	2	7	1	4	4	2
3	7	3	1	4	4	3
4	7	1	4	4	4	3
4	7	1	5	5	4	3
4	7	1	6	4	6	3
3	7	7	1	4	4	7

de donde obtenemos que el camino más corto ahora para ir de Lila a Azul es

Lila-Negra-Marrón-Roja-Naranja-Azul,

siendo la correspondiente tabla de costes

	Roja	Lila
Negra	510	20
Marrón	120!!!	410

TABLA 3. Coste de los caminos más cortos tras eliminar las esquinas Verde y Rosa

Siguiendo el esquema utilizado anteriormente, eliminamos la esquina Marrón y obtenemos el siguiente grafo

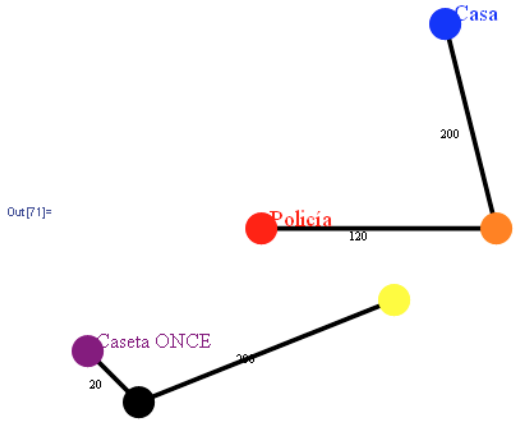

Figura 5: Grafo de la Figura 4 tras eliminar el vértice Marrón.

El conjunto de vértices eliminados del grafo original Verde, Rosa, Marrón constituye una cortadura de vértices tal que las esquinas Lila y Azul quedan en componentes conexas distintas. Por lo tanto, no existe ningún camino que en nuestro grafo actual, G-Verde, Rosa, Marrón, lleve de Lila a Azul, lo que significa que si la policía envía inmediatamente sendos agentes a las esquinas mencionadas, Pedro y Paula no tendrán un final feliz para la aventura de una noche en la que se pasaron con el alcohol.

5 Conclusiones

El ejemplo muestra la aplicabilidad de la teoría de grafos a contextos en principio ajenos a dicha teoría. Es necesario, pues, a lo largo del estudio de dicha teoría, el ver diferentes ejercicios de modelización ([1, 3, 4]), que servirán de guía para, con el tiempo y la práctica, aplicar la teoría de grafos a nuestro campo de estudio o ampliarla para que nos sea útil.

Referencias

[1] Ralph P. Grimaldi, Matemáticas discretas y combinatoria, Ed. Addison Wesley Longman (1998).

[2] J. L. Gross, J. Yellen, Graph theory and its applications, Chapman&Hall, (2006).

[3] Cristina Jordán, Juan R. Torregrosa, Introducción a la teoría de grafos y sus algoritmos, Ed. Reverté, Valencia, (1996).

[4] Kenneth Rosen, Matemática discreta y sus aplicaciones, Ed. McGrawHill, (2004).