

Grafos eulerianos y grafos hamiltonianos

Ramon Masià

Jaume Pujol

Josep Rifà

Mercè Villanueva

P06/75006/01401

Índice

| | |
|---|----|
| Introducción | 5 |
| 1. Grafos eulerianos | 7 |
| 1.1. Resultados fundamentales | 7 |
| 1.1.1. Ejercicios | 11 |
| 1.1.2. Soluciones | 12 |
| 1.2. Construcción de un circuito euleriano | 12 |
| 1.2.1. Ejercicios | 14 |
| 1.2.2. Soluciones | 14 |
| 2. Grafos hamiltonianos | 16 |
| 2.1. Resultados fundamentales | 16 |
| 2.1.1. Ejercicios | 19 |
| 2.1.2. Soluciones | 20 |
| 3. El viajante de comercio | 21 |
| 3.1. El TSP | 21 |
| 3.1.1. El TSP con desigualdad triangular | 22 |
| 3.1.2. Aproximación al TSP con desigualdad triangular | 24 |
| 3.1.3. Ejercicios | 26 |
| 3.1.4. Soluciones | 27 |
| Ejercicios de autoevaluación | 29 |
| Soluciones | 32 |
| Bibliografía | 35 |

Introducción

Muchos problemas de planificación y organización (servicio de recogida de basura, servicio de inspección de instalaciones de gas, servicio de reparto de cartas, servicios de mensajería) se pueden tratar como problemas de optimización de recorridos.

En este módulo se estudian dos problemas clásicos de optimización de recorridos: los recorridos que pasan por todas las aristas del grafo una sola vez (*recorridos y circuitos eulerianos*) y los que contienen todos los vértices del grafo una sola vez (*camino y ciclos hamiltonianos*).

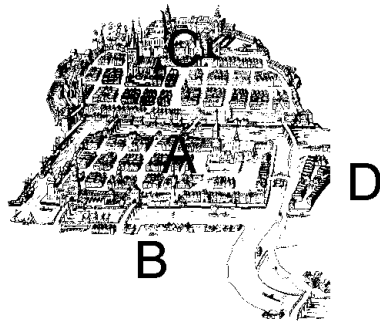
A pesar de la similitud entre los dos problemas, veremos que tienen un tratamiento diferenciado de manera que el primero es un problema “fácil” de resolver, mientras que el segundo es un problema “intratable”.

Finalmente, estudiaremos una generalización del problema de buscar un ciclo hamiltoniano en un grafo. Es el *problema del viajante de comercio (TSP)* que es uno de los problemas más notables de optimización en un grafo. Es especialmente interesante porque constituye uno de los ejemplos básicos de problema de la clase *NP*.

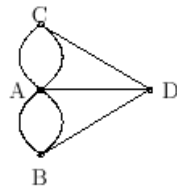
1. Grafos eulerianos

1.1. Resultados fundamentales

En el siglo XVIII la antigua ciudad de Königsberg (en Prusia oriental) estaba dividida en cuatro zonas por el río Pregel. Siete puentes comunicaban estas zonas tal y como muestra el gráfico siguiente:



Uno de los entretenimientos de los ciudadanos de Königsberg consistía en buscar un recorrido que atravesara cada puente una vez, volviendo a su punto exacto de partida. En términos de grafos este entretenimiento es equivalente a buscar un recorrido que pase por cada arista del multigrafo G



exactamente una vez.

Leonard Euler probó que no existía un recorrido que cumpliera estas condiciones y caracterizó los grafos (multigrafos) que sí lo contienen: éstos son los **grafos eulerianos**.

Definición 8.1

En un grafo (multigrafo) $G = (V, A)$

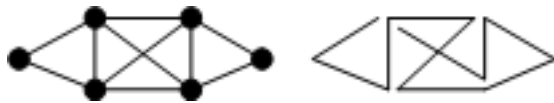
- Un **recorrido euleriano** es un recorrido abierto que contiene todas las aristas del grafo sin repetición.
- Un **circuito euleriano** es un circuito que pasa por todas las aristas del grafo. Si un grafo admite un circuito de estas características, se denomina **grafo euleriano**.

Estas nociones sólo tienen sentido en el contexto de los grafos conexos; en caso de que no lo sean, se tratan los problemas de eulerianidad en las componentes conexas del grafo.

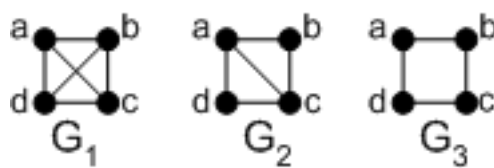
Ejemplo 8-1

De una manera intuitiva podemos decir que un grafo es euleriano si puede ser dibujado de un solo trazo sin repetir ninguna línea y empezando y acabando en un mismo punto.

El grafo de la izquierda es euleriano, puesto que lo podemos dibujar de un solo trazo sin repetir ninguna línea y empezando y acabando en un mismo punto tal como lo muestra el dibujo de la derecha.

**Ejemplo 8-2**

De los grafos representados en la siguiente figura



el grafo G_1 no contiene ningún recorrido ni circuito euleriano (obsérvese que es K_4). El grafo G_2 contiene un recorrido euleriano: a, b, c, d, a, c ; pero no contiene ningún circuito euleriano. Finalmente, el grafo G_3 contiene un circuito euleriano: a, b, c, d, a ; pero no contiene recorrido euleriano (obsérvese que se trata de C_4).

Teorema 8.2

Sea $G = (V, A)$ un grafo conexo. Entonces, G es euleriano si, y sólo si, todos los vértices son de grado par.

Demostración:

- 1) Supongamos que el grafo es euleriano y probemos que el grado de todo vértice es par. Sea Q un circuito euleriano que, en la medida que contiene todas las aristas, pasa por todos los vértices; sea v un vértice cualquiera, que tiene que pertenecer a Q , y que even-

tualmente puede estar repetido. Cada aparición del vértice en el circuito contribuye en dos unidades al grado del vértice (tenemos que contar la arista por la que “llegamos”, que se utiliza por primera vez puesto que no se pueden repetir aristas, y tenemos que “salir” por una arista aún no utilizada); la contribución es de dos unidades porque las aristas adyacentes que figuran en la secuencia no han estado utilizadas anteriormente; si k es el número de apariciones del vértice, entonces el grado es $g(v) = 2k$, es decir, par.

- 2) Supongamos que el grado de todo vértice es par y vamos a probar que el grafo es euleriano. Tenemos que encontrar un circuito euleriano en el grafo G .

Seleccionamos un vértice arbitrario v y formamos un circuito Q que se inicia en v tal como se indica a continuación. Si empezamos por v , cada vez que se alcanza un vértice se selecciona una arista que sea incidente y que no haya sido utilizada anteriormente; continuamos con el otro vértice extremo. Puesto que el grado es par, en cada vértice disponemos de una arista de salida para cada arista de llegada; la única excepción a esta regla es la del vértice inicial v , en donde se ha utilizado una arista de salida sin haber utilizado ninguna de entrada y, por lo tanto, si se llega a un vértice que ya no tiene más aristas con las que continuar el recorrido Q , este vértice es el inicial v .

Cuando esto pasa hay dos posibilidades: Q ya contiene todas las aristas del grafo, en cuyo caso ya hemos acabado y éste es el circuito euleriano que queríamos construir. No contiene todas las aristas, en este caso hemos construido un circuito en el grafo y entonces, por conexión, algún vértice del circuito es incidente con alguna otra arista no incluida en Q ; este vértice se puede utilizar como punto de partida para la construcción de un nuevo circuito Q' que no contiene aristas de Q y que devuelve al vértice de partida, puesto que en $G - Q$ cada vértice es todavía de grado par. La reunión $Q \cup Q'$ es un circuito; si contiene todas las aristas, hemos acabado. De lo contrario, se repite el proceso hasta obtener un circuito euleriano (el proceso acaba debido a la finitud del grafo).

■

Se puede enunciar un resultado más completo de caracterización de los grafos eulerianos:

Corolario 8.3

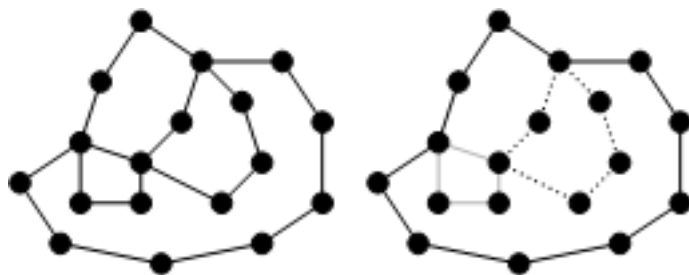
Sea $G = (V, A)$ un grafo conexo. Entonces son equivalentes:

- 1) G es euleriano.
- 2) Los grados de los vértices son pares.
- 3) El conjunto de las aristas admite una partición en circuitos que no tienen aristas en común.

Demostración: Consecuencia inmediata del teorema 8.2 de caracterización de los grafos eulerianos. ■

La tercera propiedad describe la estructura del grafo. La figura que sigue ilustra, en un caso concreto, el enunciado; el conjunto de las aristas A se puede expresar como $A = A_1 \cup \dots \cup A_k$, con $A_i \cap A_j = \emptyset$ si $i \neq j$, y el conjunto de las aristas de cada A_i forma un circuito que no comparte lógicamente ninguna arista con ningún otro de los circuitos correspondientes a los otros subconjuntos, aun cuando los circuitos mencionados pueden compartir vértices. En la primera figura se muestra el grafo original y en la otra, una descomposición

posible. La descomposición no es necesariamente única; el lector se puede dedicar a buscar otras descomposiciones posibles.

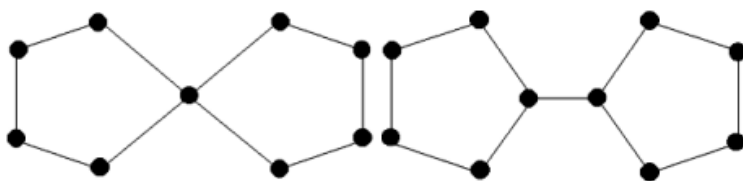


Ejemplo 8-3

A partir de la caracterización de los grafos eulerianos en términos de los grados de los vértices, resulta trivial comprobar que el (multi)grafo correspondiente al problema de los puentes de Königsberg no es euleriano y, por lo tanto, el problema no tiene solución.

Ejemplo 8-4

La primera figura corresponde a un grafo euleriano y la segunda a un grafo no euleriano. Sólo se necesita encontrar los grados de los vértices.



También nos podemos preguntar cuándo un grafo (multigrafo) conexo $G = (V, A)$ contiene un recorrido euleriano.

Corolario 8.4

Un grafo (multigrafo) conexo $G = (V, A)$ contiene un recorrido euleriano si, y sólo si, G tiene exactamente dos vértices de grado impar.

Demostración: Si G contiene un recorrido euleriano $u - v$ entonces el grado de todos los vértices de G diferentes de u y v será un número par (con el mismo razonamiento utilizado para los circuitos eulerianos) y el grado de u y v será impar.

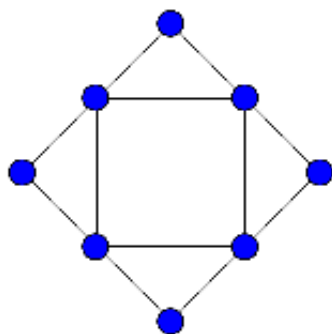
Si G contiene exactamente dos vértices u y v de grado impar, entonces el multigrafo $G + uv$ es conexo y tiene todos los vértices de grado par. Por lo tanto, si aplicamos el resultado de caracterización de los grafos eulerianos, $G + uv$ contiene un circuito euleriano C . Si eliminamos de este circuito C la arista $\{u, v\}$ obtendremos un $u - v$ recorrido euleriano en el grafo G . ■

1.1.1. Ejercicios

8-5 ¿Un circuito euleriano contiene todos los vértices del grafo?

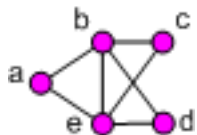
8-6 Analizar para qué valores de n, m, r, s, t son eulerianos los siguientes grafos: $K_n, R_n, E_n, C_n, K_{n,m}, K_{r,s,t}$.

8-7 ¿Cómo podemos saber si el siguiente grafo

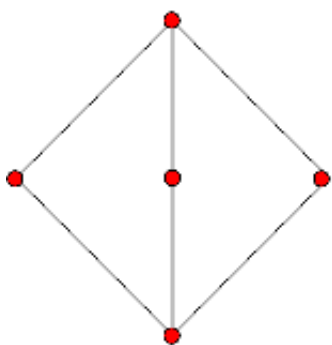


se puede dibujar de un solo trazo, sin repetir ninguna línea y empezando y acabando en el mismo vértice?

8-8 Estudiar si el siguiente grafo es euleriano utilizando la caracterización de los grafos eulerianos en términos de la descomposición del conjunto de aristas en circuitos que no tienen aristas en común.



8-9 Estudiar si el grafo siguiente es euleriano:



En caso de que no lo sea, decir si contiene un recorrido euleriano.

8-10 En una exposición se debe entrar por el mismo sitio por el que se sale. Durante un recorrido un visitante se da cuenta de que el espacio donde está tiene tres puertas. ¿Puede deducir de aquí que pasará por lo menos dos veces por la misma puerta?

8-11 ¿Cuál es el número mínimo de puentes que se necesitaría construir (y dónde) en Königsberg para que el “problema de los puentes de Königsberg” tenga solución?

1.1.2. Soluciones

8-5 Sí.

8-6 Según el teorema 8.2 (página 8) de caracterización, serán eulerianos cuando el grado de cada vértice sea par. Así, K_n será euleriano cuando n sea impar, puesto que cada vértice tiene grado $n - 1$. En R_n ($n > 3$) un vértice tiene grado $n - 1$ y el resto tienen grado 3, por lo tanto no será euleriano. En E_n ($n > 2$) hay vértices de grado 1, por lo tanto tampoco podrá ser euleriano.

En C_n todos los vértices tienen grado 2, por lo tanto será euleriano para todo n .

En $K_{n,m}$ los vértices tienen grado m o n , por lo tanto $K_{n,m}$ será euleriano cuando n y m sean pares.

Finalmente, en $K_{r,s,t}$ hay vértices de grado $s + t$, $r + t$ y $r + s$. Por lo tanto, será euleriano cuando r , s y t sean los tres pares o los tres impares.

8-7 Se podrá dibujar el grafo de un solo trazo si el grafo es euleriano, es decir, si cada vértice tiene grado par como es el caso del grafo representado por el dibujo.

8-8 El grafo es euleriano porque todos sus vértices tienen grado par. Una posible descomposición es $C_1 = \{a,b,c,e,a\}$, $C_2 = \{b,d,e,b\}$.

8-9 El grafo no es euleriano, ya que contiene vértices de grado impar. No obstante, puesto que contiene exactamente dos vértices de grado impar se puede concluir que contendrá un recorrido euleriano.

8-10 Si se considera el grafo que tiene como vértices los espacios de la exposición y, como aristas, cada una de las puertas; entonces el espacio donde se encuentra el visitante corresponde a un vértice de grado impar. Por lo tanto, el grafo no es euleriano y no se podrá construir un circuito que pase por cada arista sin repetir alguna.

8-11 Se necesitaría construir dos puentes. Por ejemplo, uno entre A y D y otro entre B y C.

1.2. Construcción de un circuito euleriano

En esta sección caracterizaremos los grafos eulerianos como aquéllos en los que el conjunto de aristas admite una partición en circuitos que no tienen aristas en común. El algoritmo resultante (primeramente atribuido a Hierholzer, 1873) construye un circuito euleriano a partir de la concatenación de circuitos disjuntos, es decir, sin aristas en común.

Formulación del algoritmo de Hierholzer

Entrada : un (multi)grafo conexo y euleriano $G = (V, A)$ de orden n
y un vértice inicial $s \in V$.

Salida : un circuito euleriano C de G representado como una lista de vértices.

algoritmo *CircuitoEuleriano*(G, s)

inicio

$C \leftarrow \{s\}$

mientras $A \neq \emptyset$

$v \leftarrow \text{VérticeGradoPositivo}(G, C)$

$C' \leftarrow \text{Circuito}(G, v)$

$C \leftarrow \text{Concatenar}(C, C', v)$

$G \leftarrow G - C'$

finmientras

retorno (C)

fin

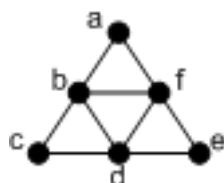
En el algoritmo se han utilizado las siguientes funciones:

- $\text{VérticeGradoPositivo}(G, C)$ devuelve el primer vértice de C que tiene grado positivo en el grafo G .
- $\text{Circuito}(G, v)$ devuelve un circuito C' construido en el grafo G a partir del vértice v .
- $\text{Concatenar}(C, C', v)$ devuelve el circuito que se obtiene de sustituir la primera aparición del vértice v en el circuito C por la totalidad del circuito C' .
- La instrucción $G \leftarrow G - C'$ elimina de G las aristas del circuito C' .

Simulación del algoritmo de Hierholzer

Ejemplo 8-12

Considerar el grafo euleriano definido por el gráfico siguiente:



Si se aplica el algoritmo de Hierholzer empezando por el vértice a se obtiene:

| Iteración | v | C' | C |
|-----------|-----|------------------|------------------------------------|
| 0 | a | | $\{a\}$ |
| 1 | a | $\{a, b, f, a\}$ | $\{a, b, f, a\}$ |
| 2 | b | $\{b, c, d, b\}$ | $\{a, b, c, d, b, f, a\}$ |
| 3 | d | $\{d, e, f, d\}$ | $\{a, b, c, d, e, f, d, b, f, a\}$ |

Así, el circuito euleriano obtenido será $C = \{a, b, c, d, e, f, d, b, f, a\}$.

Análisis del algoritmo de Hierholzer

En el algoritmo de Hierholzer se pueden distinguir las siguientes operaciones:

- 1) Elegir un vértice v de grado positivo en la lista C . Ésta es una operación lineal en la medida de C . C contendrá, como máximo, n vértices diferentes. Por lo tanto, será una operación de complejidad $O(n)$.
- 2) Construir un circuito C' en G a partir del vértice v . Ésta es una operación que depende del número de aristas elegidas. En el peor de los casos serán tantas como la medida m del grafo. Será, pues, una función de complejidad $O(m)$.
- 3) Concatenar los circuitos C y C' . Ésta es una operación lineal en la medida de C , es decir, de complejidad $O(n)$.
- 4) Eliminar de G las aristas del circuito C' . Trivialmente, es una operación que, en el peor de los casos, tendrá una complejidad $O(m)$.

Resumiendo, se puede concluir que la complejidad de todo el algoritmo será

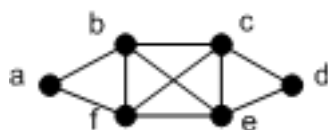
$$\max\{O(n), O(m), O(n), O(m)\} = O(m)$$

puesto que el grafo es conexo.

1.2.1. Ejercicios

8-13 ¿El circuito euleriano obtenido por el algoritmo de Hierholzer es único?

8-14 Aplicando el algoritmo de Hierholzer, encontrar un circuito euleriano del grafo



8-15 ¿Como se podría utilizar el algoritmo de Hierholzer para obtener un recorrido euleriano de un grafo G ?

1.2.2. Soluciones

8-13 No, dependerá de cómo la función *Circuito()* construye cada uno de los circuitos. En el ejemplo 8-12 (página 13) se podrían haber construido los circuitos $\{a, b, c, d, e, f, a\}$ y $\{b, d, f, b\}$, obteniendo el circuito euleriano $C = \{a, b, d, f, b, c, d, e, f, a\}$.

8-14 La siguiente tabla muestra la aplicación del algoritmo de Hierholzer empezando por el vértice a :

| Iteración | v | C' | C |
|-----------|-----|---------------------|---------------------------------------|
| 0 | a | | $\{a\}$ |
| 1 | a | $\{a, b, f, a\}$ | $\{a, b, f, a\}$ |
| 2 | b | $\{b, c, d, e, b\}$ | $\{a, b, c, d, e, b, f, a\}$ |
| 3 | c | $\{c, e, f, c\}$ | $\{a, b, c, e, f, c, d, e, b, f, a\}$ |

Así, el circuito euleriano obtenido será $C = \{a, b, c, e, f, c, d, e, b, f, a\}$.

8-15 Si el grafo G contiene un recorrido euleriano, entonces contiene exactamente dos vértices de grado impar. Sean u y v estos vértices. Se construye el grafo $G' = G + uv$ y se aplica el algoritmo de Hierholzer a G' (puesto que G' será euleriano). En el circuito obtenido sólo se necesita eliminar la arista $\{u, v\}$ para obtener el recorrido euleriano de G .

2. Grafos hamiltonianos

2.1. Resultados fundamentales

El origen de los grafos hamiltonianos está en el juego de Hamilton, en el que se debe hallar un recorrido cerrado sin repetición de vértices, a través de las aristas de un dodecaedro regular. Otros problemas como el del tablero de ajedrez, el del viajante de comercio y el del robot de soldadura controlado por ordenador han motivado el estudio de estos grafos.

Definición 8.5

En un grafo $G = (V, A)$

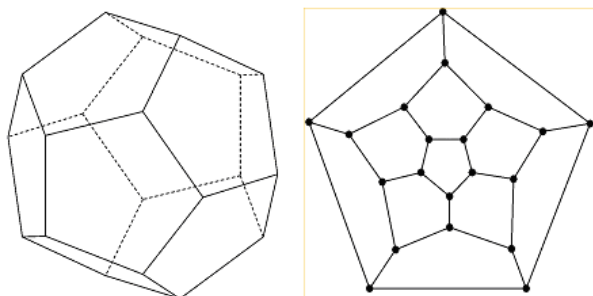
- Un recorrido es un **camino hamiltoniano** si pasa por todos los vértices sin repetición.
- Un **ciclo hamiltoniano** es un ciclo que pasa por todos los vértices del grafo. Si el grafo admite un ciclo de estas características, se denomina **grafo hamiltoniano**.

Estas definiciones sólo tienen sentido en el caso de un grafo conexo; en caso contrario, se aplican a cada una de las componentes conexas.

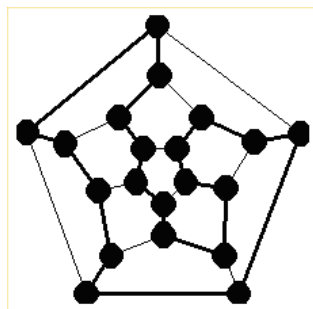
Ejemplo 8-16

El matemático W.R. Hamilton inventó en 1856 un juego denominado “*The traveller’s dodecahedron*”, que consistía en un dodecaedro cuyos vértices representaban las principales ciudades del mundo de aquella época; se tenía que encontrar un recorrido cerrado a lo largo de las aristas del poliedro que pasara por todos los vértices sin repetición.

Obsérvese que se puede considerar un modelo en términos de teoría de grafos para este entretenimiento, como se ve en el esquema adjunto: los vértices del grafo corresponden a los vértices del dodecaedro y las aristas del grafo se corresponden con las aristas del poliedro; se trata de buscar un recorrido cerrado sobre el grafo que pase por todos los vértices exactamente una vez.



En este caso, el problema tiene solución como se puede ver en el siguiente gráfico.

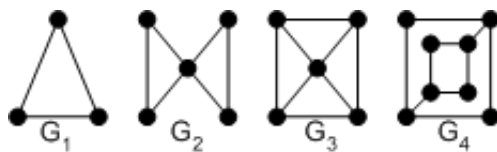


Encontrar una solución diferente de la presentada.

A pesar de la similitud entre las definiciones de grafo euleriano y grafo hamiltoniano, el siguiente ejemplo muestra que las condiciones de las definiciones son independientes:

Ejemplo 8-17

Vamos a estudiar si son eulerianos y/o hamiltonianos los siguientes grafos:



G_1 es euleriano porque todos los vértices son de grado par. También es hamiltoniano porque fácilmente se puede construir un ciclo hamiltoniano.

En G_2 todos los vértices son de grado par y, por lo tanto, es euleriano. En cambio, no es hamiltoniano. En efecto, si fuera hamiltoniano, tendría que tener un ciclo hamiltoniano, al cual contribuirían todos los vértices con exactamente dos aristas; considerando, pues, la contribución de los vértices de grado 2, resultaría que todas las aristas incidentes al vértice de grado 4 serían de este ciclo, lo que no tiene sentido.

G_3 no es euleriano (contiene vértices de grado impar) pero sí es hamiltoniano. Fácilmente se puede encontrar un ciclo hamiltoniano.

Finalmente, G_4 ni es euleriano ni hamiltoniano. No es euleriano porque contiene vértices de grado impar. Si hubiera un ciclo hamiltoniano, contendría todas las aristas de los vértices de grado 2, y puesto que cada vértice puede contener exactamente dos aristas incidentes; se tienen que descartar las aristas que conectan los vértices de grado 3 del grafo y, en consecuencia, el supuesto “ciclo hamiltoniano” sería reunión de dos ciclos!

A diferencia de los grafos eulerianos no hay ningún resultado que dé la condición necesaria y suficiente para que un grafo sea hamiltoniano. Existen condiciones necesarias de hamiltoneidad y existen también, independientemente, condiciones suficientes; muchas de éstas tienen poca aplicabilidad práctica en muchos casos concretos.

De manera intuitiva se puede observar en el ejemplo anterior que un grafo hamiltoniano es un grafo denso, es decir, contiene muchas aristas necesarias para conectar los vértices del grafo.

Primeramente, se da una definición que permitirá obtener condiciones necesarias para que un grafo sea hamiltoniano.

Definición 8.6

Un grafo $G = (V, A)$ es **2-conexo** si cada pareja de vértices u y v de G está conectada por un mínimo de dos caminos disjuntos, es decir, dos caminos que los únicos vértices que tienen en común son los extremos u y v .

Ejemplo 8-18

Si nos fijamos en los grafos del ejemplo anterior, se puede observar:

G_1 es 2-conexo, puesto que todos los vértices forman parte de un ciclo.

G_2 no es 2-conexo. En efecto, cualquier camino que una dos vértices opuestos en diagonal tiene que pasar por el vértice central.

Esta noción de 2-conectividad es necesaria para la existencia de un ciclo hamiltoniano.

Teorema 8.7

Si $G = (V, A)$ es un grafo hamiltoniano:

- 1) G es conexo y todos sus vértices tienen grado mayor o igual que 2.
- 2) G es 2-conexo.
- 3) Para todo $S \subset V$, $S \neq \emptyset$ se verifica $c(G - S) \leq |S|$ donde $c(G - S)$ representa el número de componentes conexas del grafo obtenido de G después de eliminar los vértices (y las aristas incidentes) de S .
- 4) Si G es (V_1, V_2) -bipartito entonces $|V_1| = |V_2|$.

Demostración: Si G es hamiltoniano, entonces se pueden disponer todos los vértices de G formando un ciclo $C : v_1, v_2, \dots, v_n, v_1$. Así cada vértice v_i se conecta, al menos, con v_{i-1} y v_{i+1} . Por lo tanto, $g(v_i) \geq 2$.

Del mismo modo, entre v_i y v_j ($i < j$) habrá, al menos, dos caminos disjuntos: el camino v_i, v_{i+1}, \dots, v_j y el camino $v_j, v_{j+1}, \dots, v_n, v_1, \dots, v_i$. Por lo tanto, será 2-conexo.

Si en un ciclo se eliminan r vértices, entonces se producirán como máximo r componentes conexas. Por lo tanto, para que un grafo sea hamiltoniano, siempre que se eliminen un conjunto de vértices S el número de componentes conexas producidas no puede superar el cardinal de S .

La última condición es consecuencia de la distribución de los vértices del grafo G en un ciclo. Necesariamente los vértices tienen que pertenecer alternativamente a V_1 y a V_2 . Y esto implica que $|V_1| = |V_2|$. ■

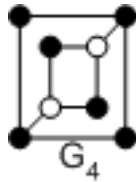
Ejemplo 8-19

Estas condiciones necesarias de hamiltoneidad se suelen utilizar para demostrar que un grafo no es hamiltoniano.

Los grafos con vértices de grado 1 no pueden ser hamiltonianos. Así, todos los grafos trayecto y todos los árboles no son grafos hamiltonianos.

El grafo G_2 del ejemplo 8-17 (página 17) no es hamiltoniano, puesto que no es 2-conexo.

El grafo G_4 del mismo ejemplo tampoco es hamiltoniano. Si se elimina el conjunto de dos vértices de color blanco se obtendrá un grafo con tres componentes conexas.



Todas estas condiciones sirven para demostrar que un grafo no es hamiltoniano, pero no permiten asegurar que lo es. Hay algunas condiciones suficientes para la existencia de ciclos hamiltonianos, pero tampoco sirven para encontrar explícitamente el ciclo. Desde un punto de vista algorítmico el problema de buscar un ciclo hamiltoniano en un grafo G es un problema NP . Es decir, un problema computacionalmente “intratable”. Esto no significa que no se puedan encontrar ciclos hamiltonianos en un grafo, pero se deben utilizar algoritmos que exploren todas las posibilidades con técnicas de vuelta atrás (*backtracking*, en inglés).

Ejemplo 8-20

Si G es un grafo de n vértices, se podrían generar todos los ciclos formados por los n vértices. Esto es equivalente a formar todas las permutaciones de n elementos que son $n!$. Naturalmente, no todas estas permutaciones dan lugar a ciclos, puesto que puede haber vértices que no son adyacentes. Además, se puede fijar un vértice para empezar siempre por el mismo sitio. También se puede fijar una orientación en el ciclo, lo que reduce el número total de permutaciones que se necesitan explorar. En total se deben explorar $\frac{(n-1)!}{2}$ posibles ciclos.

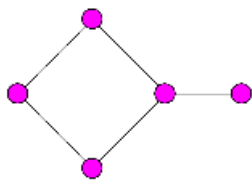
Para n pequeño, por ejemplo $n = 9$, el número de posibilidades es reducido (20160 para $n = 9$) y un computador las puede analizar en breves segundos. Sin embargo, cuando n aumenta, el gran número de posibilidades provoca que este método sea computacionalmente inviable. Por ejemplo, para $n = 20$ ya se tendrían que analizar más de 10^{17} posibilidades.

2.1.1. Ejercicios

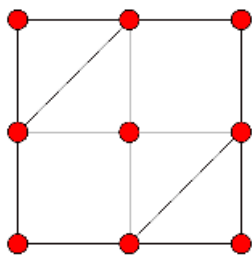
8-21 ¿Es cierto que todo grafo hamiltoniano contiene un camino hamiltoniano? ¿El recíproco es cierto?

8-22 ¿Todo ciclo hamiltoniano es circuito euleriano?

8-23 ¿Es hamiltoniano el siguiente grafo?



8-24 Estudiar si el grafo siguiente es euleriano; analizar también si es hamiltoniano.



8-25 La condición de 2-conectividad es necesaria para la hamiltoneidad. Proponer algún ejemplo que muestre que no es suficiente.

2.1.2. Soluciones

8-21 Si v_1, \dots, v_n, v_1 es un ciclo hamiltoniano en G entonces v_1, \dots, v_n es un camino hamiltoniano en G . El recíproco, no obstante, no es cierto. Por ejemplo, el grafo trayecto T_n contiene un camino hamiltoniano, pero no es un grafo hamiltoniano.

8-22 No, porque no se incluyen necesariamente todas las aristas.

8-23 No, porque contiene un vértice de grado 1.

8-24 El grafo es euleriano porque todos los vértices tienen grado par. En cambio, no es hamiltoniano. Si se considera el conjunto S formado por los cuatro vértices situados en los puntos medios de los cuatro lados, entonces $|S| = 4$, mientras que $c(G - S) = 5$ en contradicción con la condición 3 del teorema 8.7 (página 18).

8-25 El grafo del ejercicio anterior es 2-conexo, pero no es hamiltoniano. Otro ejemplo sería el grafo G_4 del ejemplo 8-17 (página 17).

3. El viajante de comercio

3.1. El TSP

Considérese un conjunto de n ciudades que un representante de una compañía de seguros tiene que visitar para dar a conocer las novedades de la empresa. Naturalmente, el representante tendrá que volver a su punto de partida. El viaje entre cada par de ciudades tiene un coste que dependerá de la distancia entre las dos ciudades. El representante se plantea qué itinerario tiene que seguir para visitar todas las ciudades con el menor coste posible.

La manera natural de enfocar el problema es considerar un grafo conexo y ponderado (G, w) donde los vértices son las ciudades y las aristas representan las distancias entre las mismas.

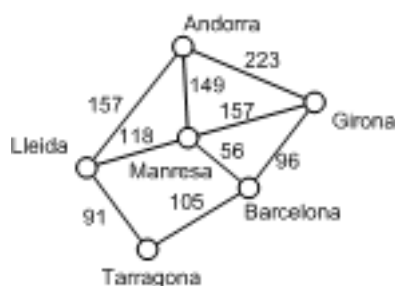
Este problema es un ejemplo del denominado **problema del viajante de comercio** (*travelling salesman problem*, en inglés) y, bajo ciertas condiciones, es equivalente al problema de buscar un ciclo hamiltoniano en el grafo G . A este último problema lo denominaremos problema *TSP*. Ahora bien, también es un problema *NP*, es decir, computacionalmente intratable. Sin embargo, veremos que en determinadas situaciones es posible aproximar la solución óptima de manera eficiente.

Formulación del problema *TSP* 8.8

Sea (G, w) un grafo conexo y ponderado, con pesos no negativos en las aristas. El problema trata de encontrar un ciclo hamiltoniano del grafo de peso (o longitud) total mínimo.

Ejemplo 8-26

Determinar la ruta óptima (que pase por cada ciudad una sola vez) que tiene que seguir el representante de la compañía de seguros en el mapa siguiente:



En este caso en que sólo hay seis ciudades podríamos mirar las $\frac{5!}{2} = 60$ posibilidades de combinar las seis ciudades (algunas combinaciones no son rutas reales), y elegir la más económica.

Si se utiliza un algoritmo *greedy* y se buscan las cinco aristas de peso más pequeño que no forman ciclo (el árbol generador minimal), se obtendrá un grafo formado por las aristas {Barcelona, Manresa}, {Lleida, Tarragona}, {Barcelona, Girona}, {Barcelona, Tarragona} y {Manresa, Andorra} con una longitud total de 497. Éste es el mínimo valor posible para la solución que se busca. En este caso, no obstante, estas aristas no forman un ciclo hamiltoniano y, por lo tanto, no son solución del problema. Son solamente una cota inferior.

En este caso, la solución óptima estaría formada por el ciclo



con una longitud total de 755, es decir, el representante recorrería 755 kilómetros como mínimo.

3.1.1. El TSP con desigualdad triangular

La mayoría de problemas reales en los cuales se tiene que buscar un ciclo hamiltoniano óptimo corresponden a situaciones en las que intervienen distancias (viajes por una red de carreteras, tiempos transcurridos, repartos de trabajos, etc.). En todas estas situaciones se verifica la *desigualdad triangular* que significa que si comparamos la distancia (coste, tiempo) entre tres puntos x, y, z siempre será más corto ir directamente de x a y que pasar por z . De manera más precisa,

Definición 8.9

Un grafo ponderado (G, w) satisface la propiedad de la **desigualdad triangular** si para toda terna de vértices de G , u, v, x ,

$$w(u, v) \leq w(u, x) + w(x, v)$$

En particular, todo grafo que satisface la desigualdad triangular tiene que ser completo.

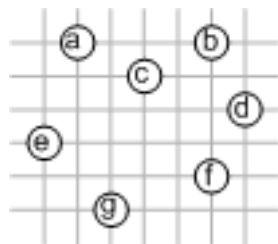
Para esta clase de grafos, el problema del viajante de comercio (*TSP*) se formularía de la siguiente manera:

Formulación del problema *TSP con desigualdad triangular* 8.10

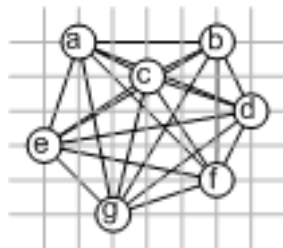
Sea (G, w) un grafo conexo y ponderado, con pesos no negativos en las aristas, que satisface la desigualdad triangular. El problema trata de encontrar un ciclo hamiltoniano del grafo de peso (o longitud) total mínimo.

Ejemplo 8-27

Un grafo ponderado que satisface la desigualdad triangular se puede representar en una porción de plano euclídeo:



Cada vértice del grafo está determinado por un par de coordenadas (v_x, v_y) y el peso de cada arista es la distancia euclídea entre los vértices. En este ejemplo se representa el grafo K_7 ponderado y que satisface la desigualdad triangular:



El vértice e está situado en la posición $(0, 2)$ y el vértice b en la posición $(5, 5)$. El peso de la arista $\{e, b\}$ es $w(e, b) = d(e, b) = \sqrt{(5-0)^2 + (5-2)^2} = \sqrt{25+9} = \sqrt{34}$.

Aunque la formulación del problema del viajante de comercio que satisface la desigualdad triangular parece más simple, continúa siendo un problema *NP*, es decir, computacionalmente intratable. Sin embargo, en la sección siguiente se desarrollará un algoritmo eficiente que permite obtener una solución aproximada del problema. En la práctica, y especialmente si el número de vértices del grafo es elevado, puede ser suficiente construir una solución aproximada del problema (tan buena como sea posible) en lugar de obtener la solución óptima. Por ejemplo, si se tiene un método para encontrar una solución de manera eficiente que no se aleje más de un 2% de la solución óptima, seguramente será una “buena solución” para la mayoría de aplicaciones prácticas (obsérvese que en aplicaciones reales –distancias, por ejemplo– puede haber errores de medición que permiten despreciar errores pequeños en la solución).

3.1.2. Aproximación al TSP con desigualdad triangular

Una forma de aproximar la solución del problema que estamos tratando consiste en considerar la manera más corta de unir todos los vértices del grafo G sin generar un ciclo. Ya sabemos que para conseguir esto tenemos que calcular el árbol generador minimal de G . La idea del algoritmo que se propone será construir un ciclo hamiltoniano en G utilizando el número máximo posible de las aristas de un árbol generador minimal de G .

Formulación del algoritmo TSP-aproximado

Entrada : un grafo conexo y ponderado (G, w) de orden n que satisface la desigualdad triangular.

Salida : un ciclo hamiltoniano H de G representado por una lista de vértices.

algoritmo $TSP\text{-}aproximado(G)$

inicio

Se selecciona $r \in V(G)$ /* la raíz del árbol */

$T \leftarrow ArbolGenerador(G, r)$

$P \leftarrow Preorden(T)$

$H \leftarrow CicloHamiltoniano(P)$

retorno (H)

fin

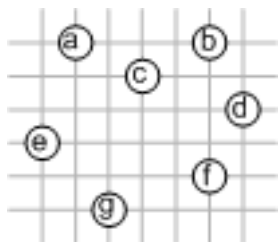
En el algoritmo se han utilizado las siguientes funciones:

- $ArbolGenerador(G, r)$ devuelve un árbol generador de G a partir de la raíz r usando el algoritmo de Prim.
- $Preorden(T)$ devuelve un recorrido en preorden del árbol con raíz T . Este recorrido contendrá todos los vértices de G en un determinado orden, $P = \{v_1, v_2, \dots, v_n\}$.
- $CicloHamiltoniano(P)$ devuelve el ciclo que se obtiene al añadir la arista $\{v_n, v_1\}$ al recorrido P , es decir, $H = \{v_1, v_2, \dots, v_n, v_1\}$.

Simulación del algoritmo TSP-aproximado

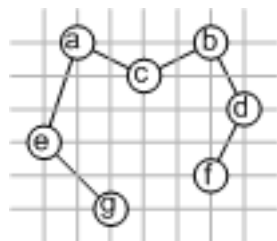
Ejemplo 8-28

Considérese el grafo del ejemplo 8-27 (página 23)



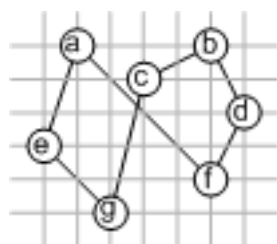
Vamos a aplicar el algoritmo *TSP-aproximado* a este grafo:

- 1) Se selecciona un vértice inicial, $a \in G$, como raíz del árbol.
- 2) Se construye el árbol generador minimal T de G empezando por a :

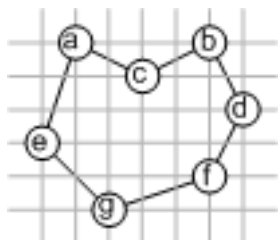


Este árbol generador minimal tiene una longitud $w(T) = 4\sqrt{5} + \sqrt{10} + \sqrt{8} = 14.935$

- 3) Se obtiene el recorrido en preorden de T : $P = \{a, e, g, c, b, d, f\}$.
- 4) Y, finalmente, el ciclo hamiltoniano $H = \{a, e, g, c, b, d, f, a\}$ que tendrá una longitud $w(H) = \sqrt{10} + \sqrt{8} + \sqrt{17} + 3\sqrt{5} + \sqrt{32} = 22.4789$



En este ejemplo, el ciclo hamiltoniano óptimo es $H^* = \{a, c, b, d, f, g, e, a\}$



con una longitud $w(H^*) = 18.0973$.

Obsérvese que un número importante de las aristas (en este ejemplo concreto, todas) del árbol generador minimal también pertenecen al ciclo hamiltoniano óptimo. También, muchas de las aristas del árbol generador minimal pertenecen al ciclo aproximado.

Análisis del algoritmo *TSP-aproximado*

El algoritmo *TSP-aproximado* efectúa dos operaciones fundamentales:

- 1) Calcula el árbol generador minimal de G con el algoritmo de Prim. Ésta es una operación que tiene una complejidad $O(n^2)$.
- 2) Calcula el recorrido en preorden de un árbol con raíz. Este algoritmo visita cada vértice una sola vez y tiene una complejidad que es una función lineal del orden del árbol, es decir, una complejidad $O(n)$.

En total, el algoritmo tendrá una complejidad $O(n^2)$.

Se puede afirmar, pues, que éste es un algoritmo eficiente que obtiene una solución aproximada al problema del *TSP con desigualdad triangular* (que es un problema intratable). Naturalmente, esta ganancia en eficiencia provoca una pérdida de precisión en la solución. No obstante, necesitaríamos asegurarnos de que esta pérdida de precisión no supera ciertos límites, es decir, la diferencia entre la solución aproximada y la óptima se mantiene acotada.

Proposición 8.11

Sea H el ciclo hamiltoniano obtenido por el algoritmo *TSP-aproximado* y denotemos por H^* el ciclo hamiltoniano óptimo (mínimo) de G . Entonces

$$w(H) \leq 2 \cdot w(H^*)$$

Demostración: Sea T el árbol generador de peso mínimo de G obtenido por el algoritmo. Entonces $w(T) \leq w(H^*)$, puesto que $w(T)$ es el peso total de unir todos los vértices de G de la manera más corta posible.

Demostraremos que $w(H) \leq 2 \cdot w(T)$ donde H es el ciclo hamiltoniano obtenido por el algoritmo *TSP-aproximado*. El recorrido en preorden de un árbol se puede entender como un recorrido en profundidad del árbol en el cual se retrocede cuando se llega a una hoja y se visita cada vértice la primera vez que se encuentra en el recorrido. El peso de un recorrido hecho de este modo será exactamente $2 \cdot w(T)$, puesto que cada arista es recorrida dos veces. Puesto que en el ciclo hamiltoniano que retorna el algoritmo se sustituye el retroceso en el árbol T por una arista directa al próximo vértice de su recorrido en preorden y puesto que G cumple la desigualdad triangular, necesariamente $w(H) \leq 2 \cdot w(T)$. De las dos desigualdades se obtiene $w(H) \leq 2 \cdot w(T) \leq 2 \cdot w(H^*)$ ■

Puesto que siempre $w(H^*) \leq w(H)$, esta proposición nos asegura que utilizando el *TSP-aproximado* podemos esperar que la solución obtenida no sea más del doble de la solución óptima. No obstante, en la práctica suele ser solamente un 15% o un 20% peor que la óptima [1].

3.1.3. Ejercicios

8-29 La solución al problema del viajante de comercio (*TSP*) formulado en 8.8 no siempre es la mejor solución para que un viajante visite todas las ciudades de su zona de representación. Supongamos que el viajante tiene que visitar cinco ciudades $\{A, B, C, D, E\}$ y que la tabla siguiente representa las distancias entre estas ciudades:

| | <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> | <i>E</i> |
|----------|----------|----------|----------|----------|----------|
| <i>A</i> | 0 | 10 | 1 | 10 | 0 |
| <i>B</i> | 10 | 0 | 1 | 0 | 10 |
| <i>C</i> | 1 | 1 | 0 | 1 | 1 |
| <i>D</i> | 10 | 0 | 1 | 0 | 10 |
| <i>E</i> | 0 | 10 | 1 | 10 | 0 |

- 1) Encontrar la solución al problema *TSP*, es decir, encontrar un ciclo hamiltoniano de peso mínimo en el grafo ponderado *G* determinado por la tabla de distancias anterior.
- 2) Demostrar que es posible encontrar un circuito sobre el grafo anterior (que pase por todas las ciudades) con un peso más pequeño que el que se ha obtenido en el apartado anterior.

8-30 Si se utiliza la misma técnica que en el ejemplo 8-26 (página 21) encontrar una cota inferior para la solución al problema *TSP* del grafo definido por la tabla

| | <i>A</i> | <i>B</i> | <i>C</i> | <i>D</i> | <i>E</i> |
|----------|----------|----------|----------|----------|----------|
| <i>A</i> | 0 | 57 | 64 | 8 | 26 |
| <i>B</i> | 57 | 0 | 88 | 54 | 34 |
| <i>C</i> | 64 | 88 | 0 | 57 | 56 |
| <i>D</i> | 8 | 54 | 57 | 0 | 23 |
| <i>E</i> | 26 | 34 | 56 | 23 | 0 |

8-31 Dar un ejemplo de un grafo en el que el algoritmo *TSP-aproximado* permita obtener la solución óptima al problema *TSP* con desigualdad triangular.

8-32 ¿Se podría utilizar el algoritmo de Kruskal en lugar del algoritmo de Prim en el algoritmo *TSP-aproximado*?

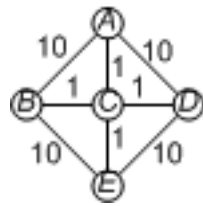
8-33 En el ejemplo 8-26 (página 21) se afirma que el número total de posibles ciclos hamiltonianos diferentes que hay en un grafo de orden n es, como máximo, $\frac{(n-1)!}{2}$. Se pospone la demostración de este resultado a un ejercicio posterior (véase el ejercicio de autoevaluación 8-38, página 29).

Hacer una tabla comparativa de los tiempos de ejecución entre un algoritmo que busque los posibles ciclos hamiltonianos (algoritmo *TSP-fuerza bruta*) y el algoritmo *TSP-aproximado* en un grafo que satisface la desigualdad triangular:

| n | <i>TSP-fuerza bruta</i> | <i>TSP-aproximado</i> |
|------|-------------------------|-----------------------|
| 5 | | |
| 10 | | |
| 50 | | |
| 100 | | |
| 500 | | |
| 1000 | | |

3.1.4. Soluciones

8-29 El grafo *G* determinado por la tabla de distancias será:



- 1) Una solución al problema *TSP* en este grafo es el ciclo $H = \{A, B, C, E, D, A\}$ con un peso total $w(H) = 32$.
- 2) No obstante, el viajante podría seguir el circuito $C = \{A, C, B, C, D, C, E, C, A\}$ con un peso total $w(C) = 8$ que es menor que el anterior. Naturalmente, en este caso se repiten vértices y aristas.

Obsérvese que este grafo no satisface la desigualdad triangular.

Esta variante del problema *TSP* se denomina problema *TSP generalizado*.

8-30 En el ejemplo 8-26 (página 21) se ha utilizado un árbol generador minimal como aproximación a la solución del problema *TSP*. Si T es el árbol generador minimal del grafo ponderado G construido a partir de la tabla del enunciado, entonces $w(T) \leq w(H^*)$ donde H^* es la solución óptima.

Si se aplican los algoritmos de Prim o de Kruskal a este grafo se obtiene un árbol T de peso $w(T) = 121$. Por lo tanto, la cota mínima de $w(H^*)$ será 121.

8-31 Por ejemplo, se puede considerar K_4 sobre el plano euclídeo con los vértices situados sobre los puntos de coordenadas $(1, 1)$, $(-1, 1)$, $(-1, -1)$ y $(1, -1)$. El peso de cada arista estaría determinado por la distancia euclídea entre los vértices que determinan la arista. En este grafo la solución dada por el algoritmo *TSP-aproximado* es óptima.

8-32 Sí. Si el árbol obtenido es el mismo, entonces la solución obtenida también será la misma.

8-33 Un algoritmo que utilice la fuerza bruta (todos los posibles ciclos hamiltonianos) para buscar el óptimo en un grafo G de orden n que satisface la desigualdad triangular tendrá que analizar $\frac{(n-1)!}{2}$ ciclos para encontrar la solución óptima. En cambio, el algoritmo *TSP-aproximado* encuentra una aproximación con un tiempo proporcional a n^2 . Si se comparan este dos valores en la tabla propuesta, se obtiene:

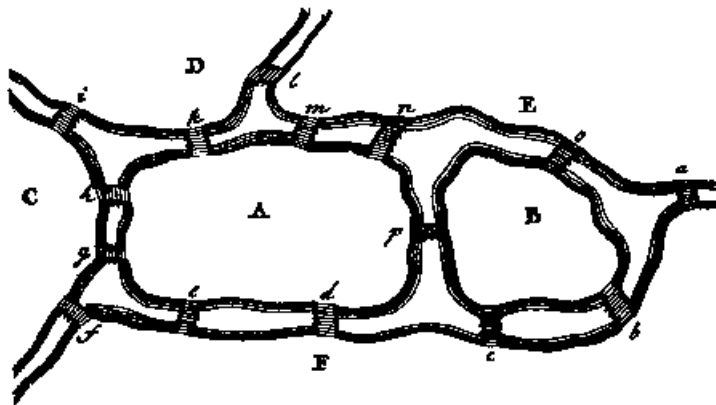
| n | <i>TSP-fuerza bruta</i> | <i>TSP-aproximado</i> |
|------|--------------------------|-----------------------|
| 5 | 12 | 25 |
| 10 | 181440 | 100 |
| 50 | 3.041×10^{62} | 2500 |
| 100 | 4.666×10^{155} | 10000 |
| 500 | 1.220×10^{1131} | 250000 |
| 1000 | 2.011×10^{2564} | 1×10^6 |

Estos resultados muestran cómo, para grafos muy pequeños, podría ser mejor utilizar un algoritmo que explore todas las posibilidades en lugar del *TSP-aproximado*. Pero, inmediatamente, cuando n aumenta, la “fuerza bruta” se vuelve intratable, mientras que el *TSP-aproximado* se mantiene dentro de unos valores aceptables.

Naturalmente, esta mejora se consigue a costa de apartarse de la solución óptima. No obstante, en muchas aplicaciones esta pérdida podría no ser significativa. En cambio el tiempo necesario para obtener la mejor solución sí que puede ser una dificultad insalvable.

Ejercicios de autoevaluación

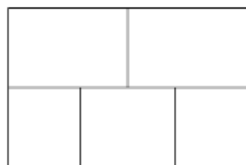
8-34 En el siguiente grafo, versión alternativa del grafo de los puentes de Königsberg, decidir si es posible hacer un recorrido cerrado que pase por todos los puentes sin repetición.



8-35 Suponiendo que un (multi)grafo conexo $G = (V, A)$ está almacenado mediante su matriz de adyacencias; ¿cómo se puede decidir fácilmente si el grafo es euleriano? ¿Qué complejidad tendrá un algoritmo que compruebe que el grafo G es euleriano? ¿Y si está almacenado mediante la lista de adyacencias?

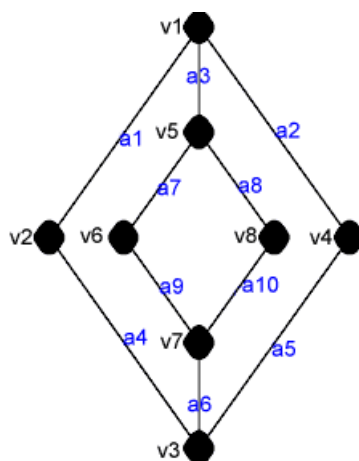
8-36 Sea $G = (V, A)$ un grafo r -regular, con $|V|$ par y $|A|$ impar. Probar que no es euleriano.

8-37 Considérese el recinto cerrado con habitaciones tal como se indica en la siguiente figura (por ejemplo, una sala de exposiciones); se abren puertas a través de las paredes que comunican habitaciones contiguas. Se trata de analizar si hay algún recorrido cerrado, que empiece y acabe en la misma habitación y que pase por todas las puertas exactamente una vez.

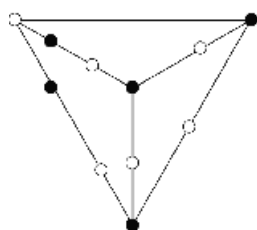


8-38 Demostrar que K_n es hamiltoniano y calcular el número de ciclos hamiltonianos que tiene, considerando que dos ciclos son diferentes si contienen por lo menos una arista diferente.

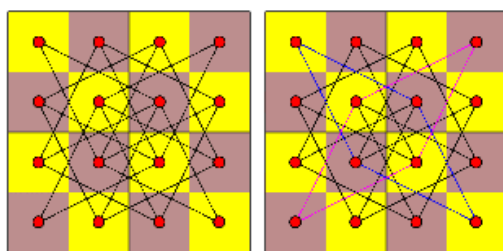
8-39 Considerar el grafo de la figura adjunta. Estudiar si es hamiltoniano por más de un método.



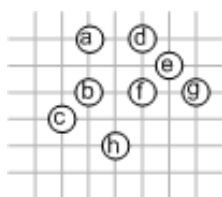
8-40 Estudiar si el grafo de la figura que sigue es hamiltoniano o no.



8-41 Probar que el problema del recorrido del caballo en un tablero de ajedrez de 4×4 no tiene solución (es decir, que el grafo correspondiente no es hamiltoniano). Estudiar también el problema con un tablero de 3×3 y uno de 5×5 . Indicación: se deben incluir necesariamente las aristas adyacentes a los vértices de grado 2.



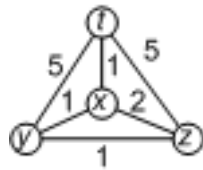
8-42 El gráfico siguiente muestra la red de puntos de reparto que una empresa de correspondencia internacional tiene que cubrir por medio de una línea aérea (cada unidad representa 1000 kilómetros):



Si el reparto empieza en el punto a y tiene que volver al mismo punto sin pasar dos veces por el mismo aeropuerto, obtener, de forma aproximada, la ruta que debe seguir y los kilómetros que tiene que hacer el avión para entregar toda la correspondencia.

8-43 Considerar el problema *TSP generalizado* en un grafo (G, w) ponderado y conexo de orden n . Se pretende buscar un recorrido cerrado de peso mínimo que pase por cada vértice (no necesariamente una única vez).

- 1) Comprobar que la solución al problema *TSP generalizado* en el grafo

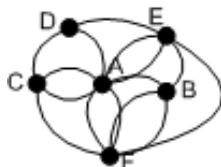


no coincide con la solución al problema *TSP*.

- 2) Proponer un método con el que se pueda resolver el problema *TSP generalizado* a partir del problema *TSP con desigualdad triangular*. Indicación: considerar el grafo de distancias de G , es decir, el grafo que se obtiene de G considerando las distancias entre todos los pares de vértices.

Soluciones

8-34 El multigrafo correspondiente a la versión alternativa de los puentes de Königsberg es



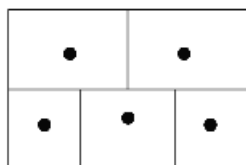
Puesto que hay vértices de grado impar (el D y el E), no es posible hacer un recorrido cerrado que pase por todos los puentes sin repetición. En cambio, puesto que hay exactamente dos vértices de grado impar, sí que se puede hacer un recorrido que pase por todos los puentes sin repetición empezando en D y acabando en E.

8-35 Si está representado por la matriz de adyacencias, el grado del vértice i -ésimo es la suma de la fila i -ésima de la matriz. Por lo tanto, el grafo será euleriano si, y sólo si, las sumas de todas las filas son números pares. Un algoritmo para comprobar si el grafo es euleriano tendría que calcular las sumas de todas las filas y, si el orden de G es n , se necesitarán un total de n^2 operaciones de suma. En definitiva, tendrá una complejidad $O(n^2)$.

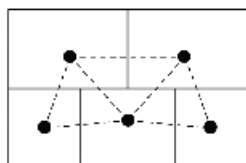
Si está almacenado como una lista de adyacencias, entonces se tendrá que verificar si cada lista tiene un número par de elementos. Si el grafo tiene orden n y medida m se necesitarán $n + 2m$ operaciones. Por lo tanto, tendrá una complejidad $O(n + m)$.

8-36 Aplicando la fórmula de los grados: $2|A| = \sum_{v \in V} g(v) = r|V|$. Si el orden del grafo es par, entonces es $|V| = 2k$, para algún k , y en consecuencia, sustituyendo en la igualdad anterior, resulta: $2|A| = 2rk$, de donde $|A| = rk$. Ahora bien, siendo impar la medida del grafo, r también es impar y, en aplicación del teorema de caracterización de grafos eulerianos (un grafo es euleriano si, y sólo si, todos los vértices son de grado par), se puede concluir que el grafo no es euleriano.

8-37 Se plantea el problema en términos de la teoría de grafos. Se asocia a cada habitación un vértice del grafo que se trata de construir, como en el siguiente esquema:



y posteriormente se representa por una arista la comunicabilidad entre habitaciones que comparten una puerta, de manera que el grafo asociado al problema es el del siguiente gráfico, donde las aristas se han indicado con línea discontinua:

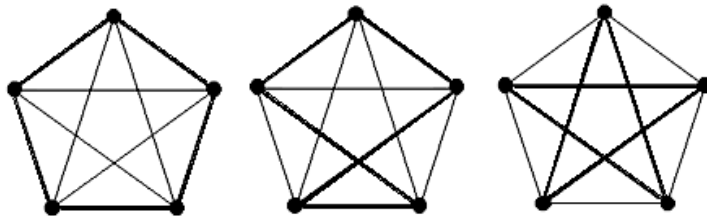


Y ahora el problema inicial se plantea como un problema de recorridos sobre el grafo asociado y, más concretamente, se pide si el grafo asociado es o no euleriano; de acuerdo con el teorema de caracterización de grafos eulerianos que dice que un grafo es euleriano si, y sólo si, todo vértice es de grado par, resulta que el grafo no es euleriano y, por lo tanto,

no es posible organizar un recorrido como el que se pide.

Obsérvese, en cambio, que puesto que hay exactamente dos vértices de grado impar, es posible organizar un recorrido que atraviese todas las puertas exactamente una vez y empiece y acabe en las habitaciones que tienen un número impar de puertas.

8-38 Puesto que todos los vértices de K_n son adyacentes, cualquier permutación de los n vértices de K_n , v_1, \dots, v_n , formarán un camino hamiltoniano. Añadiendo la arista $\{v_n, v_1\}$, se obtendrá un ciclo hamiltoniano en K_n . El gráfico siguiente muestra varios ciclos hamiltonianos en K_5 .



Para contar el número de ciclos hamiltonianos diferentes en K_n , se puede fijar un vértice v y considerar todos los ciclos que empiezan en v . En total existirán $(n-1)!$ de ellos, aunque la mitad tienen las mismas aristas con la orientación cambiada. Así, el número total de ciclos hamiltonianos diferentes en K_n es $\frac{(n-1)!}{2}$.

8-39 Método 1. Si hubiera un ciclo tendría que contener todos los vértices, lo que significa que tendría que contener todas las aristas de los vértices de grado 2; por lo tanto, se tendrían que incluir en un posible ciclo, si existiera, las aristas incidentes a los vértices de grado 2, que serían en este caso: $a_1, a_2, a_4, a_5, a_7, a_8, a_9, a_{10}$. Ahora bien, en un ciclo hamiltoniano, —puesto que es un ciclo— cada vértice sólo contribuye con exactamente dos aristas, lo que significaría que necesariamente se deben excluir las aristas a_3 y a_6 del hipotético ciclo; por consiguiente, el ciclo hamiltoniano tendría que ser finalmente $C_4 \cup C_4$, lo que es evidentemente absurdo.

Método 2. Se pretende buscar un conjunto de vértices cuya eliminación produzca un número de componentes conexas estrictamente superior al número de vértices eliminados; entonces se podrá concluir que no puede haber ningún ciclo hamiltoniano. Existe más de una posibilidad: uno de estos conjuntos de vértices estará formado por los vértices v_5, v_7 , dos vértices cuya eliminación produce tres componentes conexas; el otro conjunto estará formado por los vértices v_1, v_3 .

8-40 Éste es un ejemplo de grafo $G = (V, A)$ bipartito, como se puede ver ya directamente sobre el esquema dado: en efecto, si indicamos por V_1 el conjunto de los vértices blancos y por V_2 el conjunto de los vértices negros, entonces el grafo es (V_1, V_2) -bipartito. Ahora bien, el número de vértices negros es 5 y el número de vértices blancos es 6; en consecuencia, aplicando el criterio 4 del teorema 8.7 (página 18), el grafo no es hamiltoniano.

8-41 En términos de grafos se puede definir el problema de la siguiente manera: comprobar que el grafo $G = (V, A)$, definido por $V = \{(i, j) \mid 1 \leq i, j \leq 4\}$ donde las aristas están formadas por pares $\{(i, j), (i', j')\}$ con $|i - i'| = 1$ y $|j - j'| = 2$ o $|i - i'| = 2$ y $|j - j'| = 1$, es o no hamiltoniano.

Si el grafo fuera hamiltoniano, las aristas incidentes a los cuatro vértices de las esquinas, que tienen grado 2, deberían estar en el ciclo. Estas aristas determinan dos ciclos disjuntos (véase el gráfico), lo que impide que formen parte del mismo ciclo hamiltoniano.

8-43 Este ejercicio relaciona tres problemas aparentemente diferentes: *TSP*, *TSP generalizado* y *TSP con desigualdad triangular*.

- 1) La solución al problema *TSP* para este grafo sería el ciclo $H_1 = \{t, x, y, z, t\}$ con una longitud total de 8. En cambio, la solución al problema *TSP generalizado* será el circuito $H_2 = \{t, x, y, z, x, t\}$ con una longitud de 6. Observemos que este grafo no satisface la desigualdad triangular.
- 2) Considérese el grafo de distancias del grafo G . Este grafo será K_4 con el mismo conjunto de vértices de G y con todas las aristas posibles. El peso de la arista $\{u, v\}$ será $w(u, v) = d(u, v)$ en G . Se puede calcular la matriz $D = (d_{i,j})$ de distancias del grafo G a partir de su matriz de pesos $W = (w_{i,j})$ aplicando el algoritmo de Floyd.

A partir de la matriz W de pesos:

| | x | y | z | t |
|-----|-----|-----|-----|-----|
| x | 0 | 1 | 2 | 1 |
| y | 1 | 0 | 1 | 5 |
| z | 2 | 1 | 0 | 5 |
| t | 1 | 5 | 5 | 0 |

se obtiene la matriz de distancias D aplicando el algoritmo de Floyd:

| | x | y | z | t |
|-----|-----|-----|-----|-----|
| x | 0 | 1 | 2 | 1 |
| y | 1 | 0 | 1 | 2 |
| z | 2 | 1 | 0 | 3 |
| t | 1 | 2 | 3 | 0 |

El grafo ponderado (K_4, d) obtenido a partir de la matriz D es un grafo que satisface la desigualdad triangular. Por lo tanto, se puede buscar la solución al problema *TSP con desigualdad triangular* en este grafo: $H_3 = \{t, y, z, x, t\}$ con una longitud igual a 6.

Este ejercicio muestra que es posible reducir el problema *TSP generalizado* al problema *TSP con desigualdad triangular* y que la solución a éste da una solución al primero. También muestra que es tan difícil resolver el primero como el segundo, es decir, junto con el *TSP* son problemas *NP*.

Bibliografía

1. Skiena, Steven S. (1998). *The Algorithm Design Manual*. Berlín: Springer-Verlag.

