Aula 11 – Trilhas e Circuitos

Notas de Aula de Teoria dos Grafos

Prof[a.]: Patrícia D. L. Machado

UFCG – Unidade Acadêmica de Sistemas e Computação

Sumário

Nesta aula, estudaremos o conceito de trilhas e circuitos (de Euler), o problema do carteiro chinês (*chinese postman problem*) e revisitaremos o problema das 7 pontes de Könibsberg.

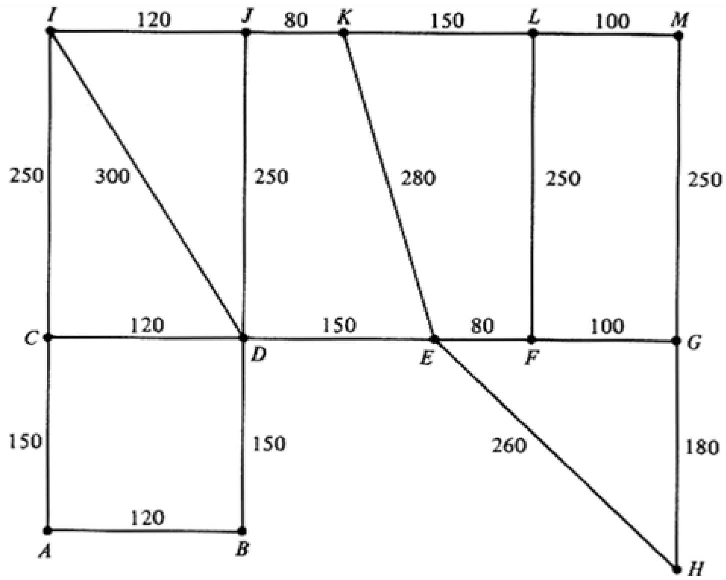
Chinese Postman Problem

O problema do carteiro chinês (*Chinese Postman Problem*) consiste no seguinte. Suponha que um carteiro precisa percorrer as ruas de uma vizinhança para entregar correspondências usando a menor rota possível e considerando os seguintes critérios:

- A rota é um circuito fechado (inicia e termina no mesmo ponto)
- Ele deve passar por cada rua pelo menos 1 vez.

Como podemos representar e resolver este problema usando grafos? Inicialmente precisamos representar o domínio em termos de um grafo, onde os vértices representam os cruzamentos das ruas e as ruas são representadas por arestas. Podemos definir um grafo ponderado para então representar o comprimento de cada trecho de uma rua como sendo o peso da aresta correspondente. Abaixo vemos uma possível representação para uma vizinhança que um carteiro precisa visitar[1].

[1] Figura de: http://web.mit.edu/urban_or_book/www/book/chapter6/6.4.1.html



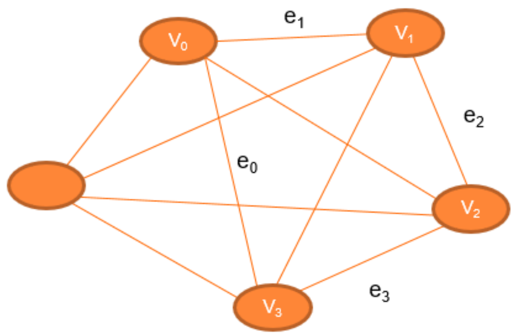
Neste caso, o problema de encontrar a melhor rota pode ser agora resolvido com um algoritmo que defina a menor rota fechada neste grafo que percorre todas as arestas (trechos de rua).

Podemos encontrar, em Ciência da computação, problemas semelhantes ao do carteiro chinês e que podem ser resolvidos através de algoritmos que calculam a menor rota (ou menores rotas):

- Teste de Sistema. Seleção de casos de teste que percorra todos os eventos de entrada e saída em um sistema (vértices representam estados e arestas eventos de entrada e saída)
- Roteamento e gerenciamento em redes de computadores.

Conectividade

Estamos em aulas anteriores o conceito de passeio. Este conceito pode ser utilizado para definirmos a conectividade em grafos. Se existe um passeio *W* que inicia em um vértice *x* e termina em um vértice *y,* dizemos que *W* **conecta** *x* a *y* ou *W* é um *xy*-passeio ou W é um *x*-passeio. Lembrando que um passeio em um grafo é **fechado (*closed*)** se um mesmo vértice é o vértice inicial e terminal. No grafo abaixo, $W_0 = (v_0e_1v_1, v_1e_2v_2, v_2e_3v_3)$ é um passeio que *conecta* v_0 a v_3. Observe também que $W_1 = (v_0e_1v_1, v_1e_2v_2, v_2e_3v_3, v_3e_0v_0)$ é um passeio fechado.

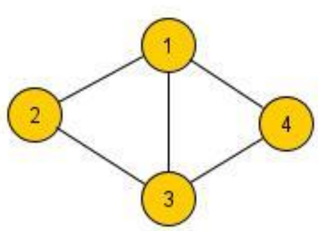
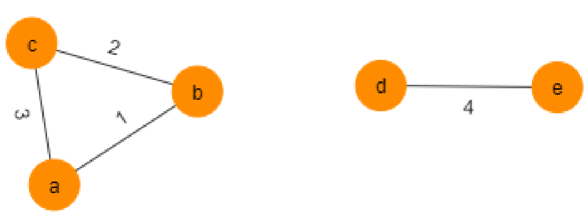


Esta noção de conectividade usando o conceito de passeio em grafos não-direcionados define uma **relação de equivalência** no conjunto de vértices que é:

- **Reflexiva** – todo vértice *x* está conectado com ele mesmo (passeio trivial $W = x$);
- **Simétrica** – Se *x* está conectado a *y* pelo passeio *W*, então *y* está conectado a *x* pelo passeio obtido da sequência inversa;
- **Transitiva** – Se *x* está conectado a *y* pelo passeio W_1 e *y* está conectado a *z* pelo passeio W_2, então *x* está conectado a *z* pelo passeio W_1W_2.

Note que, quando o grafo é direcionado, não temos esta mesma relação de equivalência, visto que não podemos garantir a simetria. Estudaremos conectividade em grafos direcionados em aulas posteriores.

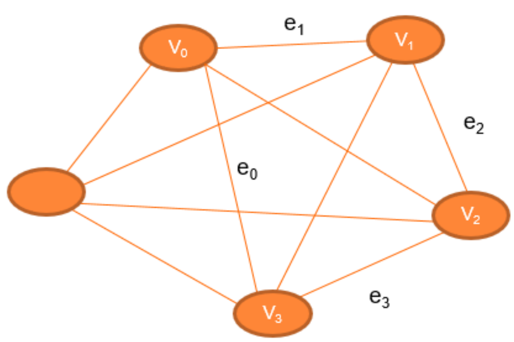
Um **componente** *C* de um grafo é um conjunto formado por vértices que são mutuamente conectados. As classes de equivalência são formadas pelos conjuntos de vértices dos componentes de *G.* Como podemos observar nos exemplos abaixo, quando o grafo é conectado, este possui um único componente, visto que todos os seus vértices são mutuamente adjacentes. O mesmo não ocorre com grafos que não são conectados.

GRAFO	COMPONENTE(s)
	$C_1 = \{1,2,3,4\}$
	$C_1 = \{a,b,c\}$ $C_2 = \{d,e\}$

A classe **[ConnectivityInspector]** da **JGraphT** possui métodos que determinam se um grafo é conectado (*isConnected*), se existe um passeio entre dois vértices (*pathExists*), quais os componentes de um grafo (*connectedSets*) e quais os vértices que pertencem ao mesmo componente de um vértice do grafo (*connectedSetOf*).

Trilhas e Circuitos

Um passeio em um grafo é uma **trilha (*trail*)** se todas as suas arestas são distintas. Como podemos observar no exemplo abaixo, $W_1 = (v_0e_1v_1, v_1e_2v_2, v_2e_3v_3)$ e $W_2 = (v_0e_1v_1, v_1e_2v_2, v_2e_3v_3, v_3e_0v_0)$ são exemplos de trilha, onde W_2 é uma trilha fechada. Porém, $W_3 = (v_0e_1v_1, v_1e_2v_2, v_2e_3v_3, v_3e_0v_0, v_0e_1v_1, v_1e_4v_3)$ não é uma trilha visto que repete a restas e_1.



Note que ***caminho e trilha são conceitos diferentes***. Em um caminho simples, não é admitida a repetição de vértices. Assim, o passeio $W_1 = (v_0e_4v_2, v_2e_2v_1, v_1e_1v_0, v_0e_0v_3)$ é uma trilha, mas não é um caminho simples.

Uma trilha que passa por todas as arestas de um grafo é chamada de **trilha de Euler (*Euler trail*)**. Note que uma trilha possui arestas distintas, mas não precisa passar por todas as arestas.

Um **circuito** (***circuit/tour***) em um grafo conectado é um passeio fechado sem arestas repetidas (uma trilha fechada). Note que um ciclo simples não permite a repetição de vértices, mas vértices podem ser repetidos em um circuito.

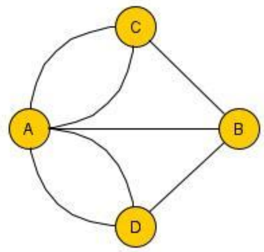
Um **circuito de Euler (*Euler Tour*)** é aquele que atravessa todas as arestas do grafo exatamente uma vez (uma trilha de Euler fechada)

Um grafo é **Euleriano** se admite um circuito de Euler.

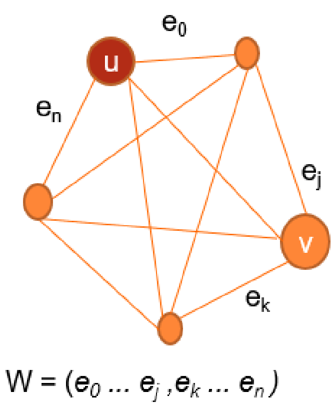
Mostrar que o **problema das 7 pontes de *Königsberg*[2]** que estudamos anteriormente não tem uma solução se resume a mostrar que o grafo correspondente (abaixo ilustrado) não possui uma trilha ou circuito de Euler.

---

[2] https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg



Vamos agora observar a propriedades básicas que um grafo Euleriano deve exibir. Seja *G* um grafo Euleriano e *W* um circuito de Euler de *G*, sendo *u* o vértice inicial e terminal (conforme exemplo abaixo)

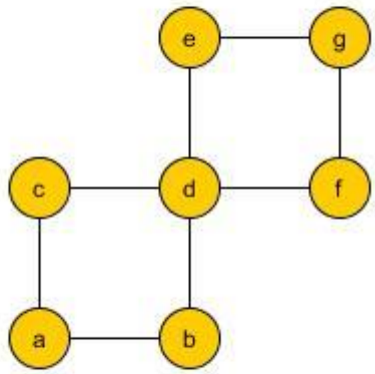


Observe que sempre que um vértice *v* ocorre como um vértice interno de *W*, duas arestas incidentes são consideradas (veja no exemplo as arestas e_j e e_k). Visto que um circuito de Euler atravessa uma aresta exatamente uma vez, podemos concluir que o grau de *v*, *d(v)*, é par para todo $v \neq u$. De forma análoga, podemos concluir que o grau de *u*, *d(u)*, é par, visto que *W* começa e termina em *u*. Desta forma, um grafo Euleriano é necessariamente par.

Teorema. Um grafo conectado é **Euleriano** sss é par.

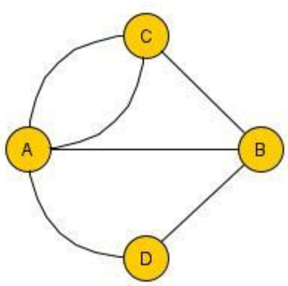
Assim, podemos concluir que o problema das 7 pontes de *Königsberg* não tem solução já que o grafo correspondente não é par, ou seja, não é possível definir um circuito de Euler – um passeio que saia de um ponto, passe por todas as pontes exatamente uma vez retornando ao ponto de partida.

Abaixo temos um exemplo de um grafo Euleriano. Um circuito de Euler pode ser definido como: (g, e, d, c, a, b, d, f, g)



Uma **trilha aberta de Euler** não exige que o vértice inicial seja igual ao final. A condição essencial para que um grafo possua uma trilha aberta de Euler é que o grafo possua exatamente 2 vértices de grau ímpar. Estes serão os vértices final e inicial do passeio: por serem diferentes, precisam necessariamente ter grau ímpar, pois a trilha parte de um e não retorna. Chega ao final mesmo sem ter passado anteriormente e não prossegue.

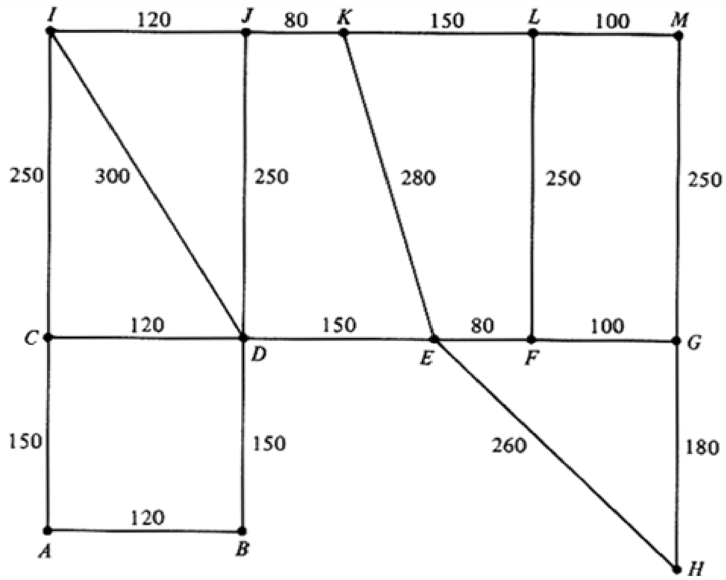
Como podemos observar, o grafo das 7 pontes também não admite uma trilha aberta de Euler, visto que possui 4 vértices de grau ímpar. Porém, considere a remoção de uma de suas arestas. Neste caso, passamos a ter exatamente 2 vértices de grau ímpar, como podemos observar no exemplo abaixo. Agora poderemos definir uma trilha de Euler de B para C ou de C para B, sendo estes os vértices de grau ímpar e os únicos que podem ser origem ou destino da trilha.



Chinese Postman Problem (Revisitando)

Revisitando o exemplo do problema do carteiro chinês que vimos no início desta aula, não é possível neste grafo encontrar um circuito de Euler para ser percorrido pelo carteiro, assim ele necessariamente precisará percorrer um ou mais trechos de rua mais de uma vez. No entanto, podemos ainda encontrar uma rota que seja a menor possível mesmo com tais repetições.

Algoritmos têm sido investigados na prática para encontrar tais rotas minimizando a repetição de arestas e buscando a rota de menor peso. A JGraphT possui uma implementação destes algoritmos na classe **ChinesePostman.** Abaixo temos o grafo que representa o exemplo e um rota saindo e retornando ao vértice H.



O algoritmo da JGraphT retorna a seguinte rota de peso 3890:

(H, G, F, G, M, L, K, E, F, L, K, J, I, D, B, D, C, A, C, I, J, D, E, H)

O grafo pode ser representado através da seguinte matriz.

,I,J,K,L,M,C,D,E,F,G,A,B,H
I,,120,,,,250,300,,,,,,
J,,,80,,,,250,,,,,,
K,,,,150,,,,280,,,,,
L,,,,,100,,,,250,,,,
M,,,,,,,,,,250,,,
C,,,,,,,120,,,,150,,
D,,,,,,,,150,,,,150,
E,,,,,,,,,80,,,,260
F,,,,,,,,,,100,,,
G,,,,,,,,,,,,,180
A,,,,,,,,,,,,120,
B,,,,,,,,,,,,,
H,,,,,,,,,,,,,

Exercícios Resolvidos

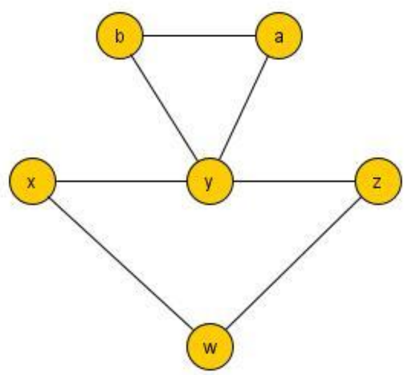
É verdade que todo grafo completo de 3 ou mais vértices possui um circuito de Euler?

Não. Apenas os grafos pares possuem circuito de Euler. Existem grafos completos que não são pares, a exemplo do K_4.

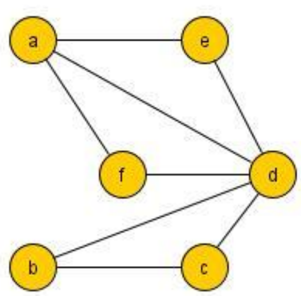
Sejam xy e yz duas arestas de um grafo conexo G sem vértices de grau ímpar. É verdade que G tem um ciclo euleriano na qual xy e yz aparecem consecutivamente?

Isto é verdade para muitos grafos, devido a paridade dos graus dos vértices. Mas considere o grafo abaixo. Para este grafo, é possível definir um circuito de Euler, mas *xy* e *yz* não podem aparecer consecutivamente em nenhum deles, considerando qualquer vértice como origem.

V(G) = {a,b,x,y,z,w}; E(G) = {ab,ay,by,xy,yz,zw,wx}



Para o grafo abaixo, encontre um circuito ou trilha de Euler se existir.



Como este grafo não é par, não é possível encontrar um circuito de Euler. Porém, como possui apenas dois vértices de grau ímpar (*a* e *d*), então é possível encontrar uma trilha de Euler. Um exemplo seria: (d,e,a,d,c,b,d,f,a).

Exercícios Propostos[3]

1. Se existe um uma *xy*-trilha em G, mostre que também existe um *xy*-caminho.

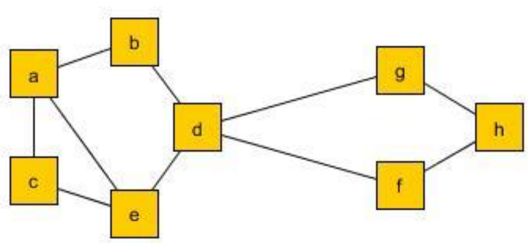
2. Mostre que um grafo *G* é conectado se e somente se existe um *XY*-caminho em G (de um vértice de X para um vértice de Y) para quaisquer subconjuntos não-vazios de *V*(G).

3. Mostre que se *e* pertence a E(G), então c(G\e) = c(G) ou c(G\e)=c(G)+1 (onde c(G) é o número de componentes)

4. Se for possível, mostre um exemplo de um grafo Euleriano com número par de vértices e número ímpar de arestas. Caso contrário, explique porque este grafo não existe.

5. Prove que, para qualquer par (x, y) de vértices de qualquer grafo, vale uma e apenas uma das seguintes afirmações: (1) um caminho liga x a y ou (2) um corte vazio separa x de y. (Outra maneira de formular a mesma questão: prove que existe um caminho de x a y se e somente se nenhum corte vazio separa x de y.)

6. Encontre um circuito ou trilha de Euler no grafo abaixo, se existir.



7. Seja (*v0, v1, v2, . . . , vk*) um ciclo em um grafo G. Mostre que há um circuito no subgrafo ({*v1, v2, . . . , vk*}, {*v0v1, v1v2, . . . , vk−1vk*}) de G.

8. Suponha que (*v0, . . . , vk*) é um passeio fechado em um grafo G. É verdade que G tem um circuito?

Referências

J. A. Bondy and U. S. R. Murty. Graph Theory. Springer, 2008,2010.

3.1, 3.3, 3.4

https://en.wikipedia.org/wiki/Route_inspection_problem

https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg

[3] Algumas questões foram adaptadas de: http://www.ime.usp.br/~pf/grafos-exercicios/