O problema do carteiro chinês

1 As sete pontes de Königsberg

O problema das sete pontes de Königsberg foi descrito e solucionado pelo matemático Leonhard Euler em 1736, o problema consistia em decidir se seria possível traçar no mapa de Königsberg um trajeto que percorresse cada uma de suas 7 pontes uma única vez, sem repetições.

A importância deste problema é enorme para a área da computação, já que o artigo de Euler publicado sobre ele é considerado o primeiro artigo sobre teoria dos grafos.

Vamos definir como **passeio** em um grafo uma sequência finita não vazia $P = \{v_0, a_1, v_2, a_2, \ldots, a_k, v_k\}$, cujos termos são alternadamente vértices v_i e arestas a_j e tal que, para todo i, $1 \leq i \leq k$, os extremos de a_i são v_{i-1} e v_i. Os vértices v_0 e v_k são a origem e o término de P, respectivamente; e os vértices $v_1, v_2, \ldots, v_{k-1}$ são chamados vértices internos de P.

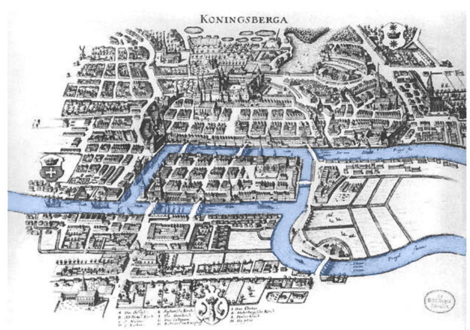

Figura 1: Representação das sete pontes de Königsberg

Uma **trilha** é um passeio sem arestas repetidas. Um **caminho** é um passeio sem vértices repetidos. Definimos o comprimento de um passeio, denotado por $||P||$, como o número de arestas de P.

Um passeio é considerado **fechado** se sua origem e término são iguais.

Uma trilha fechada, cuja origem e vértices internos são todos distintos, é um **circuito**.

Devido às contribuições de Euler ao problema descrito, chama-se **trilha euleriana** como uma trilha que passa por todas arestas de um grafo e **grafo euleriano** como um grafo que possui uma trilha euleriana fechada.

Euler modelou essa área da cidade como um grafo, representado na figura 2, tratando as pontes como arestas e as massas de terra como vértices. A partir de tal modelagem e das definições feitas, o problema de Königsberg consiste, em definir se o grafo que representa a cidade possui ou não uma trilha euleriana.

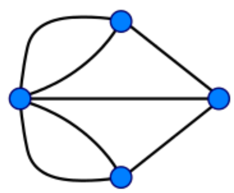

Figura 2: Modelagem de Königsberg como um grafo

O matemático provou que uma condição necessária para que um grafo seja euleriano é que todos seus vértices possuam grau par; ele também afirmou, porém sem prova, que qualquer grafo conexo cujos vértices possuem grau par é euleriano.

A prova para essa afirmação só foi publicada mais de cem anos depois, em 1873, por Carl Hierholzer [2].

Teorema 1.1 (Teorema de Euler ou de Euler-Hierholzer)**.** *Um grafo é euleriano se, e somente se, é conexo e todos seus vértices possuem grau par.*

Antes de mostrar a prova de tal teorema, primeiramente apresentaremos o seguinte lema:

Seja $\delta(G)$ o grau mínimo de um vértice pertencente a G.

Lema 1. *Se G é um grafo tal que $\delta(G) \geq 2$, então G possui um caminho fechada.*

Demonstração. Vamos assumir que um grafo qualquer $G = \{V, A\}$ não possui um caminho fechado. Seja P um caminho de comprimento maximal pertencente a G, denominamos v um dos extremos de P. Como P é um caminho de comprimento maximal, é impossível, por definição, que v possua uma aresta vu que o ligue a um vértice u não pertencente a P.

Como, da premissa, todos vértices possuem grau maior ou igual a 2, isso implica que v possuirá ao menos duas arestas que o ligam à vértices pertencentes a P.

Porém, como v é um vértice extremo de P, apenas uma dessas arestas pode pertencer ao caminho P. Isso implica que a outra aresta, digamos vw, não pertencente a P, implica na existência de um caminho fechado.

Basta tomar o subcaminho entre os vértices v e w pertencente à P juntamente com a aresta vw que teremos um caminho fechado. Chegando assim em uma contradição.

Demonstra-se assim que G deverá possuir ao menos um caminho fechado dadas as condições do lema. $\square$

Provado tal lema, podemos agora provar o teorema 1.1:

Demonstração. Seja $G = \{V, E\}$ o grafo em questão.

Começamos provando que se um grafo é euleriano e conexo então todos seus vértices possuem grau par.

Seja T uma trilha euleriana fechada, cuja existência é garantida já que G é euleriano. Cada vez que um vértice v ocorre em T ela é precedida de uma aresta que liga um vértice qualquer a v e sucedida de outra aresta ligando v a um vértice qualquer. Um caso especial é quando o vértice v em questão é a origem (e o término) de T: neste caso a aresta que o precede é a última aresta presente em T, e a aresta que o sucede é a primeira aresta da trilha.

Agora, provaremos por indução no número de arestas de G que se G for conexo e se todos seus nós possuem grau par, então ele é euleriano.

O caso base da indução é quando não há arestas em G. O único grafo conexo que respeita tal condição é o grafo que possui apenas um vértice v. Neste exemplo, $\{v\}$ é a trilha euleriana fechada do grafo.

A hipótese de indução é que todo grafo simples, conexo, que possui até $k-1$ arestas e cujos vértices tem grau par é euleriano. Seja G um grafo conexo, de vértices de grau par e que possua k arestas.

Como G é conexo, vale que $\delta(G) \geq 1$, mas como todos nós de G têm grau par, podemos afirmar que $\delta(G) \geq 2$. Sendo assim, pelo lema 1, G deverá possuir um caminho fechado C.

Se C possui todas arestas de G, então C é uma trilha euleriana fechada do grafo, finalizando a prova.

Do contrário, retira-se de G as arestas pertencentes a C, resultando assim em um grafo G'. Possivelmente G' será desconexo, por isso dizemos que G' será igual a união de componenetes conexas $G'_1, G'_2, G'_3, \ldots, G'_k$ disjuntas entre si.

O grau dos vértices dessas componenentes G'_i deverá ser par, já que ao retirar todas as arestas de uma trilha fechada do grafo G estamos diminuindo os graus de seus vértices em uma quantidade par, mantendo assim a paridade dos graus.

Além disso, cada componenete conexa de G' possuirá uma quantidade de arestas menor do que k, sendo assim, pela hipótese da indução, cada uma dessas componentes deverá possuir uma trilha euleriana fechada própria. Chamaremos de T_i a trilha euleriana fechada da componente G'_i.

Agora, construiremos uma trilha euleriana fechada para G:

Começamos a construção de nossa trilha euleriana fechada a partir de um vértice qualquer de C.

□

Corolário 1. *Um grafo possui um caminho euleriano se, e somente se, é conexo e possui apenas zero ou dois vértices de grau ímpar.*

Demonstração. Seja G um grafo conexo qualquer. Realizaremos a demonstração para os seguintes casos:

1. G não possui vértices de grau ímpar. Neste caso, G possui, segundo o teorema 1.1, um circuito euleriano, que é um tipo de caminho euleriano.

2. G possui apenas um vértice de grau ímpar. Este caso é impossível de se acontecer, já que a soma do grau de todos vértices deve ser par.

3. G possui dois vértices de grau ímpar. Sejam u e v os únicos vértices de G que possuem grau ímpar. Adiciona-se uma aresta fictícia ao grafo G, a aresta uv, fazendo com que tanto u quanto v possuam graus pares. Chamaremos o grafo G acrescido da aresta uv de G'. Como u e v eram os únicos vértices de grau ímpar de G, vale que todos vértices de G' possuirão grau par. Além disso, vale que G' é conexo, pois faz parte da premissa que o grafo original G era conexo.

Sendo assim, podemos aplicar o teorema 1.1, provando a existência de um circuito euleriano em G', que chamaremos de C. Como C é um circuito euleriano, isso implica que $uv \in C$.

4. G possui três ou mais vértices de grau ímpar.

$\square$

Como os quatro vértices do grafo da figura 2 possuem grau ímpar, é impossível, pela condição estabelecida, que exista um caminho euleriano.

Segue agora uma implementação de um algoritmo que encontra o circuito euleriano de um grafo dado que o mesmo possui um circuito euleriano:

```
// adjacency list
vector<pii> adj[MAXN];

int mrk[MAXE];

int temp_path[MAXE], final_path[MAXE], tsz, fsz;

int edge_id = 0;
void add_edge(int a, int b){
        // undirected edge
        adj[a].pb(pii(b, edge_id));
        adj[b].pb(pii(a, edge_id));
        edge_id = edge_id + 1;
}

void euler_dfs(int u){
        temp_path[tsz++] = u;
        for(pii ar: adj[u]){
                int id = ar.snd;
                if(mrk[id]) continue;
                mrk[id] = 1;
                int v = ar.fst; euler_dfs(v);
        }
        final_path[fsz++] = temp_path[--tsz];
}
// Cycle stored in final_path[0...fsz-1]
```

2 Undirected CPP

O grafo deverá ser conexo.

Solução:

1. Criar um grafo completo ligando todos vértices u, v que possuirem grau ímpar com arestas de custo igual ao menor caminho entre u e v.
2. Rodar algoritmo de emparelhamento perfeito de custo mínimo (Húngaro com otimização do Edmonds e Karp, $\mathcal{O}(n^3)$).

Observações:

- Pelo "Lema do aperto de mão" é garantido que haverá um número par de vértices de grau ímpar.
- Essa solução não se aproveita do grafo ser esparso, há outra formulação do Edmonds e Johnson que leva isso em consideração.
- Um problema similar é o de cobrir todas arestas com ciclos simples, de modo que o comprimento total dos ciclos é minimizado. Para grafos planares esses problemas são equivalentes.

3 Directed CPP

O grafo deverá ser fortemente conexo.

Solução:

1. Criar um grafo P, Q-bipartido completo. P deve conter todos os vértices do grafo original com excesso de grau de entrada, e Q deve conter todos vértices com excesso de grau de saída. que possuem valores diferentes de grau de entrada e saída. O custo das arestas entre P e Q deverá ser igual ao custo do menor caminho entre os dois vértices que a mesma liga.
2. Modelar uma rede de fluxo: vértices de P recebem um excesso igual a diferença do grau de entrada pelo grau de saída, vértices de Q recebem uma demanda igual a diferença do grau de saída pelo grau de entrada.
3. Rodar um algoritmo de fluxo de custo mínimo.

Observações:

- Modelagem de fluxo com arestas de capacidade unitária, aumentando a velocidade do algoritmo.

4 Mixed CPP

NP-hard, mesmo se G for planar e se todos c_{ij} forem iguais.

5 Windy Postman Problem

NP-hard

Se todo ciclo do grafo tem o mesmo custo em ambos sentidos, transforma a aresta entre i e j de custos c_{ij} e c_{ji} em uma aresta não direcionada com custo $\frac{c_{ij}+c_{ji}}{2}$. Essa transformação reduz o problema de a um CPP não direcionado, que tem solução polinomial, porém para isso é necessário checar se todo ciclo tem o mesmo custo nas duas direções.

6 Hierarchical Postman Problem

NP-hard

Seja $P = \{A_1, A_2, \ldots, A_k\}$ uma partição do conjunto de arestas A. Determinar um caminho em G de custo mínimo que sai de um vértice s e atinge um vértice t, respeitando uma hierarquia das arestas. O caminho só poderá possuir uma aresta pertencente a A_j se o caminho passar anteriormente por todas arestas pertencentes a A_i, se $i < j$.

7 Anotações

- Todo mixed CPP pode ser transformado em um WPP.

Referências

[1] Euler, Leonhard *Solution problematis ad geometriam situs pertinentis*. Comment. Acad. Sci. U. Petrop 8, 128–40, 1736.

[2] Hierholzer, Carl *"Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren"*, Mathematische Annalen, 6 (1): 30–32, doi:10.1007/BF01442866, 1873.