BCC204 - Teoria dos Grafos

Marco Antonio M. Carvalho

(baseado nas notas de aula do prof. Haroldo Gambini Santos)
Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

6 de novembro de 2019

UFOP
Universidade Federal de Ouro Preto

decom
departamento de computação

Avisos

Site da disciplina:

- `http://www.decom.ufop.br/marco/`

Lista de e-mails:

- `bcc204@googlegroups.com`

Para solicitar acesso:

- `http://groups.google.com/group/bcc204`

Conteúdo

O Problema do Carteiro Chinês

Introdução

Considere serviços como coleta de lixo ou correios. Os cruzamentos das ruas são vértices do grafo e as arestas são as ruas. Cada rua tem um custo de percurso associado que representa a distância, tempo ou outro fator.

É necessário percorrer **todas as ruas** e retornar ao **ponto inicial** com custo mínimo.

Aplicações

- Coleta de lixo;
- Vendas em domicílio;
- Entrega do correio;
- Recenseamento;
- Nebulização contra dengue; etc...

O Problema do Carteiro Chinês

Histórico

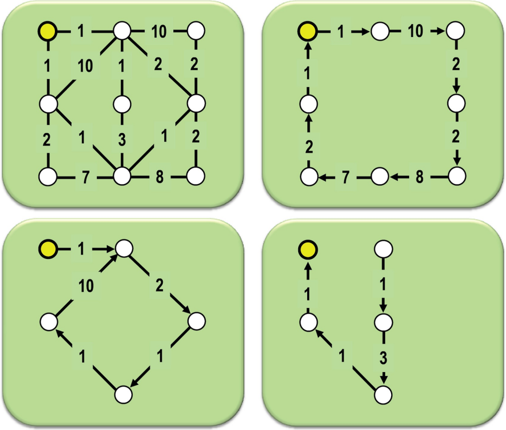
A literatura relata um grande número de problemas de otimização combinatória associados aos percursos desenvolvidos sobre as arestas de um grafo G.

O Problema do Carteiro Chinês foi relatado inicialmente por Kwan Mei-Ko em 1962, na revista *Chinese Mathematics*.

Definição

O Problema do Carteiro Chinês – PCC, consiste em determinar um passeio fechado de **custo mínimo** que passe por cada aresta de um grafo G conectado pelo menos uma vez.

O Problema do Carteiro Chinês



Grafo de exemplo e dificuldade em resolver o problema em grafos não eulerianos.

Complexidade

PCC em Grafos Não Orientados

Para o caso de grafos não orientados, a solução exata deste problema pode ser obtida em $O(n^3)$[a], portanto, em tempo polinomial.

[a] Papadimitriou & Steiglitz (1982)

PCC em Grafos Orientados

Para o caso de grafos orientados, a solução exata deste problema pode ser obtida utilizando um algoritmo de fluxo em redes, portanto, em tempo polinomial.

PCC em Grafos Mistos

Para o caso de grafos que possuem ambos os tipos de arestas, orientadas e não orientadas, o problema é NP-Difícil.

PCC em Grafos – Caso Simétrico

Para o caso de grafos nos quais as distâncias são iguais independente do sentido de travessia, o problema possui solução em tempo polinomial.

PCC em Grafos – Caso Íngreme

Para o caso de grafos nos quais as distâncias dependem do sentido de travessia, o problema é NP-Difícil.

Solução

Em grafos eulerianos e não orientados, a solução consiste em determinar o ciclo euleriano.

A solução deverá consistir em um *itinerário único*, de modo que caso o grafo não seja euleriano, algumas arestas serão percorridas *mais de uma vez*.

O processo de solução, no caso de trabalharmos com um grafo não euleriano, adiciona arestas até que se obtenha um grafo euleriano. Desta forma, indicamos quais arestas serão percorridas duas vezes.

Abordagem para Grafos Não Orientados e Conexos

1. Verifique se G é euleriano;
2. Caso positivo vá para 5;
3. Caso negativo, o grafo possui vértices de grau ímpar;
4. Adicione arestas ao grafo, duplicando as arestas que formam o caminho mais curto entre os vértices de grau ímpar, de modo que se tornem vértices de grau par;
5. Aplique um algoritmo de determinação de ciclos eulerianos.

Hierholzer e Fleury

A determinação da composição de ciclos eulerianos pode ser realizada em tempo determinístico polinomial.

Estudaremos dois agoritmos, que partem do princípio que o grafo é euleriano, ou seja, obedecem ao Teorema de Euler.

Princípio

O algoritmo de Hierholzer, proposto em 1873, foi um dos primeiros a tratar ciclos eulerianos.

A idéia é, a partir de um vértice qualquer, percorrer arestas até retornar ao vértice inicial. Porém, desta forma, pode ser obtido um ciclo que não inclua todas as arestas do grafo.

Enquanto houver um vértice que possui arestas ainda não exploradas, comece um caminho neste vértice e tente voltar a ele, usando somente arestas ainda não percorridas.

Utiliza o conceito de grafo reduzido, ou seja, remove arestas e vértices do grafo original à medida em que os insere na solução.

Terminologia

- C: conjunto das arestas que definem um ciclo euleriano no grafo;
- A_1: conjunto de arestas do grafo G ainda não percorridas;
- K: grafo reduzido criado a partir de G, porém, $K = (V, A_1)$;
- H: conjunto de arestas que definem um ciclo no grafo K;
- $\setminus$: subtração de conjuntos;
- $\cup$: união de conjuntos.

Algoritmo de Hierholzer

Entrada: Grafo $G = (V, A)$

Escolha qualquer vértice $v \in V$;

Construa um ciclo C a partir do vértice v, percorrendo as arestas de G de maneira aleatória;

$A_1 \leftarrow A \setminus C$;

$K \leftarrow (V, A_1)$;

enquanto $A_1 \neq \emptyset$ **faça**

> **Escolha** um vértice v tal que $d(v) > 0$ e $v \in C$;
>
> **Construa** um ciclo H a partir do vértice v, percorrendo as arestas de K de maneira aleatória;
>
> $A_1 \leftarrow A_1 \setminus H$;
>
> $C \leftarrow H \cup C$;
>
> $H \leftarrow \emptyset$;

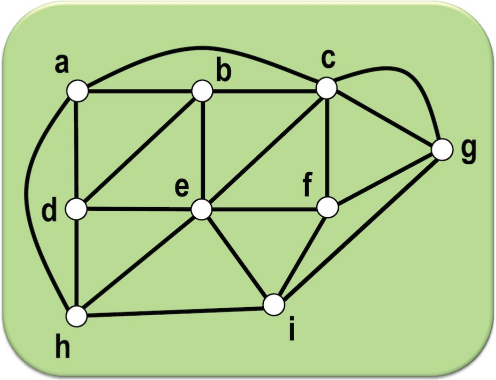
fim

Complexidade

O algoritmo de Hierholzer pode ser implementado em $O(m)$ caso sejam utilizadas listas duplamente encadeadas para:

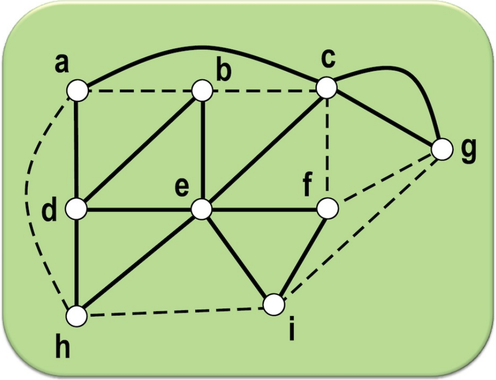
- Implementar a lista de adjacências de cada vértice;
- Implementar os ciclos C e H;
- Implementar uma lista L que contém os vértices de C com grau maior que zero no grafo reduzido.

Exemplo



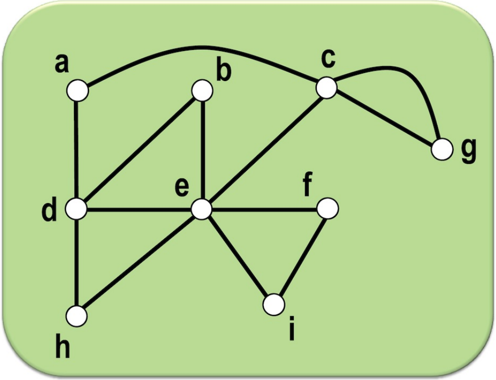
Grafo G.

Exemplo



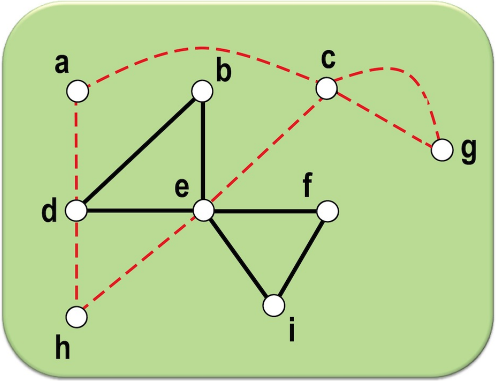
$$C = \{c, f, g, i, h, a, b\}.$$

Exemplo



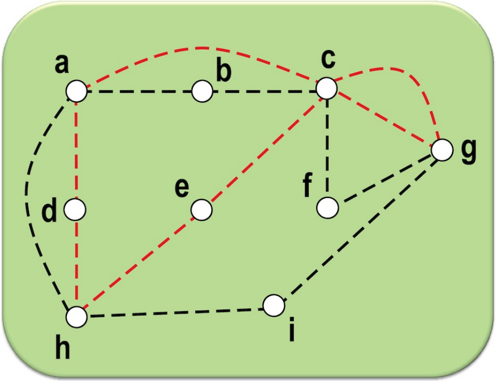
Grafo K na primeira iteração.

Exemplo



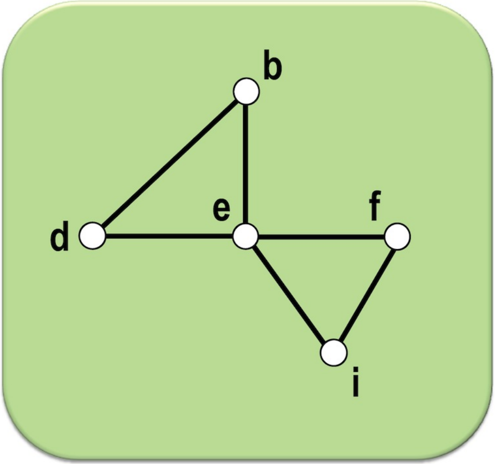
$$H = \{a, c, g, c, e, h, d\}.$$

Exemplo



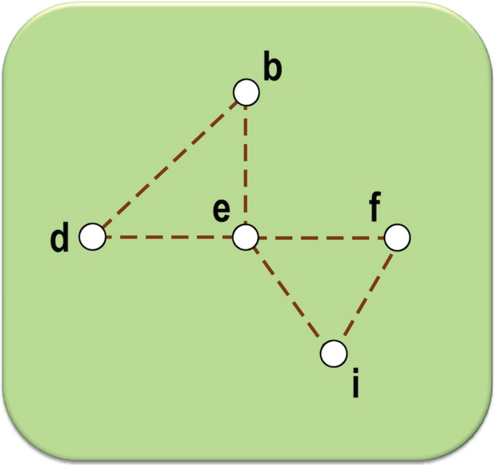
$$C = \{c, f, g, i, h, a, c, g, c, e, h, d, a, b\}.$$

Exemplo

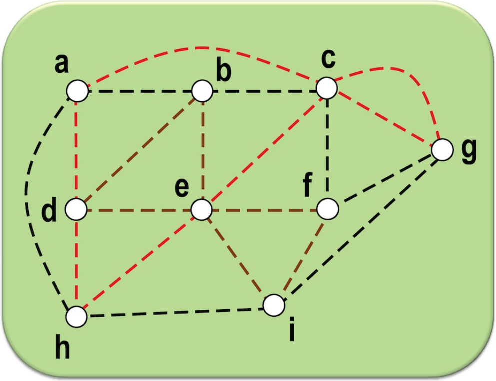


Grafo K na segunda iteração.

Exemplo



$$H = \{d, b, e, f, i, e\}.$$



$$C = \{c, f, g, i, h, a, c, g, c, e, h, d, b, e, f, i, e, d, a, b, c\}.$$

Algoritmo de Fleury

Princípio

O algoritmo de Fleury, proposto em 1883, também utiliza um grafo reduzido induzido pelas arestas ainda não marcadas pelo algoritmo.

Inicialmente todas as arestas estão não marcadas, e, a partir de um vértice aleatório, uma aresta que obedeça a regra da ponte é escolhida para ser percorrida e inserida no ciclo.

Regra da Ponte

Se uma aresta $\{v, w\}$ é uma ponte no grafo reduzido, então $\{v, w\}$ só deve ser escolhida pelo algoritmo de Fleury caso não haja outra opção.

Terminologia

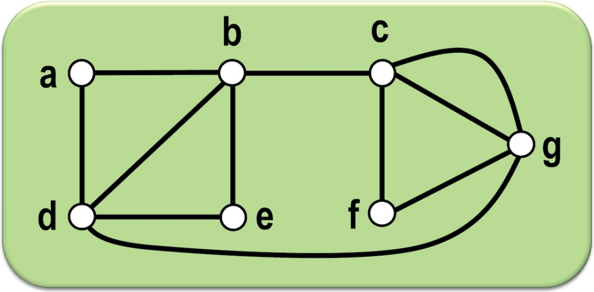
- C: conjunto das arestas que definem um ciclo euleriano no grafo.

Entrada: Grafo $G = (V, A)$

Escolha qualquer vértice $v \in V$;
$C \leftarrow \{v\}$;
repita
    Escolha uma aresta $\{v, w\}$ não marcada usando a regra da ponte;
    Atravessar $\{v, w\}$;
    $C \leftarrow C \cup \{w\}$;
    Marcar $\{v, w\}$;
    $v \leftarrow w$;
até *que todas as arestas estejam marcadas*;
$C \leftarrow C \cup \{v\}$;

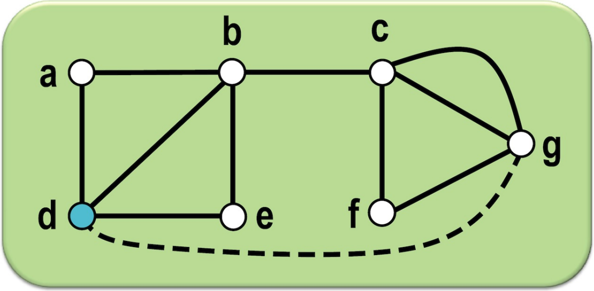
Complexidade

- $O(m)$ remoções de arestas;
- $O(m)$ para detecção de pontes (usando um algoritmo ingênuo);
- Resultando em complexidade $O(m^2)$.



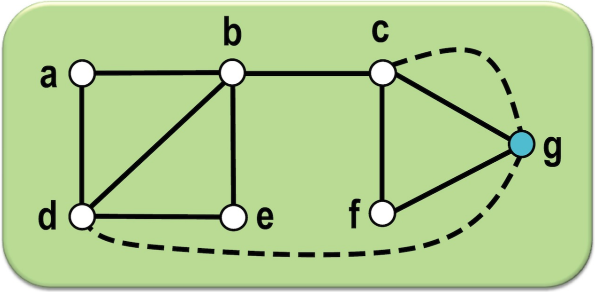
Grafo G.

Exemplo

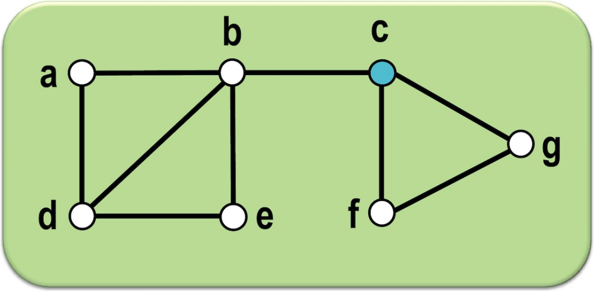


$$(v, w) = (d, g).$$

Exemplo

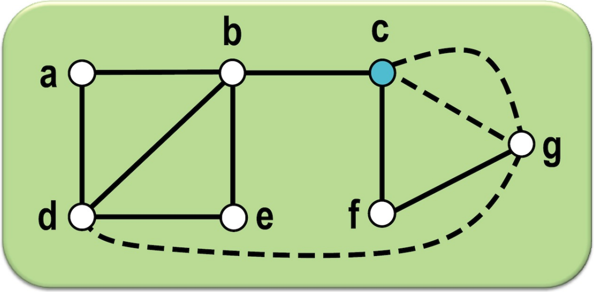


$$(v, w) = (g, c).$$



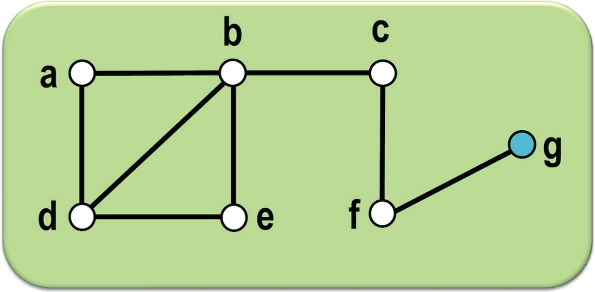
Grafo reduzido na terceira iteração.

Exemplo

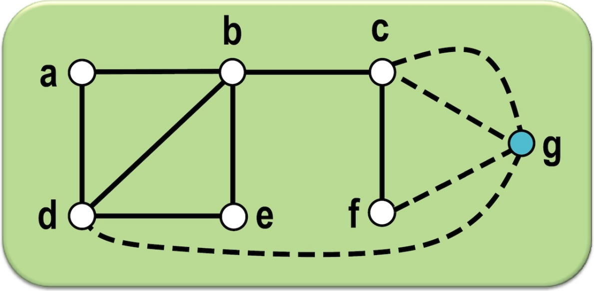


$$(v, w) = (c, g).$$

Exemplo

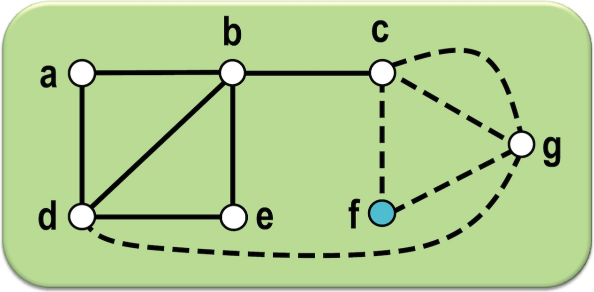


Grafo reduzido na quarta iteração.



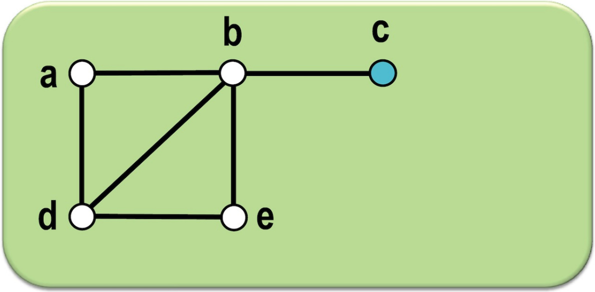
$(v, w) = (g, f)$.

Exemplo



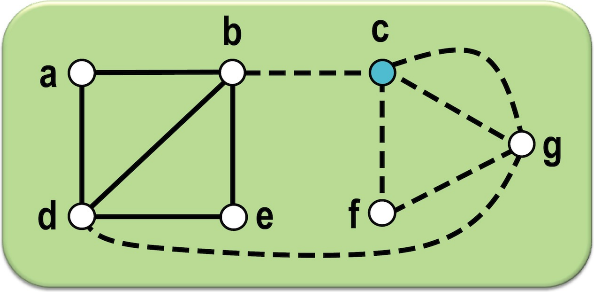
$$(v, w) = (f, c).$$

Exemplo



Grafo reduzido na sexta iteração.

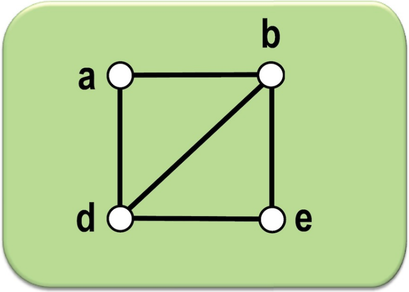
Exemplo



$$(v, w) = (c, b).$$

Exercício

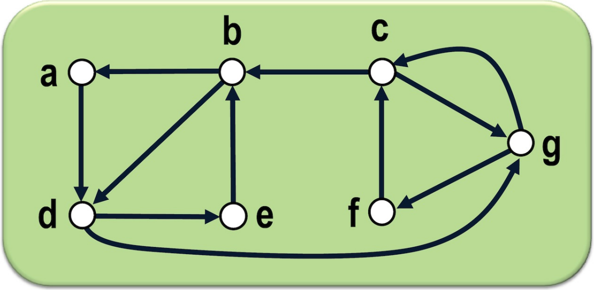
Dado o grafo reduzido, termine a execução do algoritmo.



Grafo reduzido.

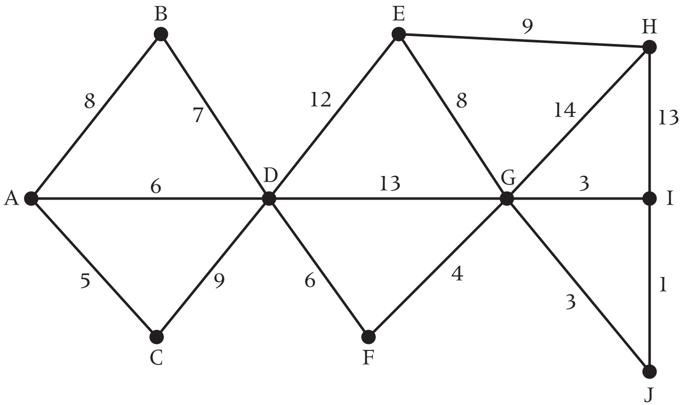
$C = \{d, g, c, g, f, c, b\}$

Exemplo



$$C = \{d, g, c, g, f, c, b, a, d, e, b, d\}.$$

Exercício



Torne o grafo acima euleriano e determine a solução do PCC.

Dúvidas?