Notas de Aula
Fundamentos de Grafos

Obs.: Estas notas de aula NÃO dispensam a consulta de referências complementares, citadas no plano de ensino.

Material organizado pelo Prof. Ricardo Moraes

SUMÁRIO

1 - Teoria dos Grafos 4
1.1 Introdução 4
2. Conceitos Básicos da Teoria dos Grafos 5
2.1 Grafo 5
2.2 Dígrafos (Grafo Orientado) 5
2.3 Ordem 6
2.4 Adjacência 6
2.5 Grau 6
2.6 Fonte 7
2.7 Sumidouro 7
2.8 Laço 7
2.9 Grafo Regular 7
2.10 Grafo Completo 7
2.11 Grafo Bipartido 8
2.12 Grafo Rotulado 8
2.13 Grafo Valorado 8
2.14 Multigrafo 9
2.15 Subgrafo 9
2.16 Hipergrafo 9
2.17 Nomenclatura e Notação 10
Exercícios 11
2.18 Mais Alguns Conceitos 13
2.18.1 Cadeia 13
2.18.2 Caminho 13
2.18.3 Ciclo 13
2.18.4 Circuito 13
2.18.5 Fecho Transitivo 13
3 Conexidade e Distância 15
3.1 Tipos de Conexidade 15
3.2 Teorema de f-conexidade 16
3.3 Grafo Reduzido 17
3.4 Critérios de Conexidade 18

3.5 Algoritmos para decomposição de um grafo em componentes f-conexas ____ *18*
3.5.1 Algoritmo de Malgrange ____ 18
3.5.2 Outra forma de encontrar as componentes conexas ____ 21
3.6 Base e Anti-base ____ *22*
Definição 1 ____ *22*
Definição 2 ____ *22*
3.6.1 Teorema 1 ____ 24
3.6.2 Teorema 2 ____ 24
3.7 Algoritmo ____ *24*
4. Distância ____ *28*
4.1 Excentricidade (afastamento) ____ *28*
4.2 Raio ____ *28*
4.3 Centro ____ *28*
4.4 Localização do Centro de Emergência ____ *28*
4.5 Teorema Sobre Matrizes Booleanas de adjacência e Alcançabilidade ____ *30*
5. Caminhos ____ *31*
5.1 Problema do Caminho de Custo Mínimo ____ *31*
5.2 Grafos Eulerianos ____ *32*
5.2.1 Problema das Pontes de Königsberg ____ 33
5.2.2 Problema do desenho da casa ____ 33
5.3 Grafos Hamiltonianos ____ *34*
5.3.1 O Caso Count Van Diamond ____ 34
5.3.2 Problema do Caixeiro Viajante ____ 35
5.3.3 Método Algébrico ____ 36
5.4 Menor Caminho Entre Dois Vértices ____ *37*
5.4.1 Algoritmo de Floyd ____ 37
5.4.2 Algoritmo de Dijkstra (1950) ____ 38
Idéia Geral do Algoritmo de Dijkstra ____ *38*
5.5 Matriz de Roteamento ____ *39*
5.5.1 Algoritmo de Floyd Modificado ____ 39

1 - Teoria dos Grafos

1.1 Introdução

Muitos problemas envolvendo o inter-relacionamento de elementos, em diversas áreas tais como: química orgânica, eletricidade, organização, transporte, psicosociologia e etc, podem ser estudados com o auxílio de grafos - face à existência de associações ou de correspondências entre os elementos em um determinado problema (átomos em uma molécula, componentes em um circuito elétrico, funcionários em uma empresa, estradas unindo cidades, crianças em uma turma de escola, etc).

A teoria está também intimamente relacionada com muitos ramos da matemática, incluindo teoria de grupos, teoria de matrizes, análise numérica, probabilidade, topologia e combinatória. O fato é que a teoria dos grafos serve como um modelo matemático para qualquer sistema que envolva uma relação binária.

HISTÓRICO

O primeiro problema (historicamente documentado) que se preocupou com a representação da relação das componentes de um problema é conhecido como o "Problema das Pontes de Königsberg".

Euler conseguiu demonstrar que de qualquer ilha para se percorrer todas as pontes que ligavam as duas costas da cidade, seria necessário passar por alguma ponte mais de uma vez (Figura 1). No entanto, sua demonstração apenas pode ser compreendida após representar a situação por meio de um grafo (esboçado na Figura 2).

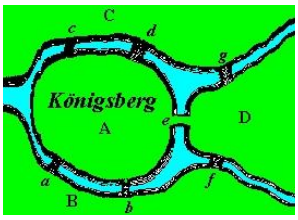


Figura 1 - "Problema das Pontes de Königsberg".

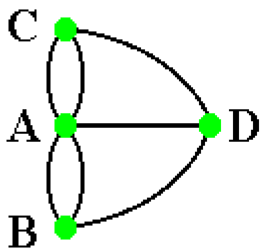


Figura 2 - Grafo alusivo ao problema da Figura 1.

2. Conceitos Básicos da Teoria dos Grafos

2.1 Grafo

Um grafo G(V, A) é definido pelo par de conjuntos V e A, onde:

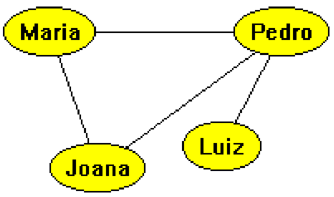
V - conjunto não vazio: os vértices ou nodos do grafo;
A - conjunto de pares ordenados a=(v,w), v e w ∈ V: as arestas do grafo.

Seja, por exemplo, o grafo G(V,A) dado por:

V = { p | p é uma pessoa }
A = { (v,w) | < v é amigo de w > }

Esta definição representa toda uma família de grafos. Um exemplo de elemento desta família (ver G1) é dado por:

V = { Maria, Pedro, Joana, Luiz }
A = { (Maria, Pedro) , (Joana, Maria) , (Pedro, Luiz) , (Joana, Pedro) }



G1

Neste exemplo estamos considerando que a relação <v é amigo de w> é uma relação simétrica, ou seja, se <v é amigo de w> então <w é amigo de v>. Como conseqüência, as arestas que ligam os vértices não possuem qualquer orientação.

2.2 Dígrafos (Grafo Orientado)

Considere, agora, o grafo definido por:

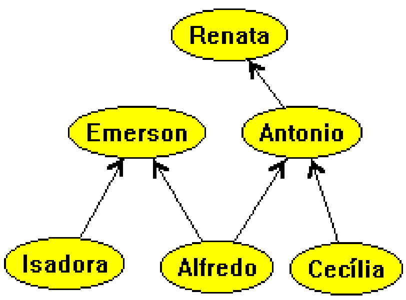
V = { p | p é uma pessoa da família Castro }

A = { (v,w) | < v é pai/mãe de w > }

Um exemplo de deste grafo (ver G2) é:

V = { Emerson, Isadora, Renata, Antonio, Rosane, Cecília, Alfredo }

A = {(Isadora, Emerson), (Antonio, Renata), (Alfredo, Emerson), (Cecília, Antonio), (Alfredo, Antonio)}



G2

A relação definida por A não é simétrica, ou seja, se <v é pai/mãe de w>, não significa que <w é pai/mãe de v>. Há, portanto, uma orientação na relação, com um correspondente efeito na representação gráfica de G.

O grafo acima é dito ser um grafo orientado (ou dígrafo), sendo que as conexões entre os vértices são chamadas de arcos.

2.3 Ordem

A ordem de um grafo G é dada pela cardinalidade do conjunto de vértices, ou seja, pelo número de vértices de G. Nos exemplos anteriores:

ordem(G1) = 4
ordem(G2) = 6

2.4 Adjacência

Em um grafo simples (por exemplo G1) dois vértices v e w são adjacentes (ou vizinhos) se há uma aresta a=(v,w) em G. Está aresta é dita ser incidente a ambos, v e w. É o caso dos vértices Maria e Pedro em G1. No caso do grafo ser dirigido (a exemplo de G2), a adjacência (vizinhança) é especializada em:

Sucessor: um vértice w é sucessor de v se há um arco que parte de v e chega em w. Em G2, por exemplo, diz-se que Emerson e Antonio são sucessores de Alfredo.

Antecessor: um vértice v é antecessor de w se há um arco que parte de v e chega em w. Em G2, por exemplo, diz-se que Alfredo e Cecília são antecessores de Antonio.

2.5 Grau

O grau de um vértice é dado pelo número de arestas que lhe são incidentes. Em G1, por exemplo:

grau(Pedro) = 3
grau(Maria) = 2

No caso do grafo ser dirigido (a exemplo de G2), a noção de grau é especializada em:

2.5.1 Grau de emissão: o grau de emissão de um vértice v corresponde ao número de arcos que partem de v. Em G2, por exemplo:

GrauDeEmissão(Antonio) = 1
GrauDeEmissão(Alfredo) = 2
GrauDeEmissão(Renata) = 0

2.5.2 Grau de recepção: o grau de recepção de um vértice v corresponde ao número de arcos que chegam a v. Em G2, por exemplo:

GrauDeRecepção(Antonio) = 2
GrauDeRecepção(Alfredo) = 0
GrauDeRecepção(Renata) = 1

2.6 Fonte

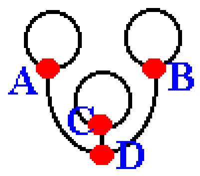
Um vértice v é uma fonte se GrauDeRecepção(v) = 0. É o caso dos vértices Isadora, Alfredo e Cecília em G2.

2.7 Sumidouro

Um vértice v é um sumidouro se GrauDeEmissão(v) = 0. É o caso dos vértices Renata e Emerson em G2.

2.8 Laço

Um laço é uma aresta ou arco do tipo a=(v,v), ou seja, que relaciona um vértice a ele próprio. Em G3 há três ocorrências de laços para um grafo não orientado.

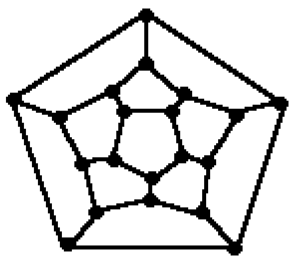


G3

2.9 Grafo Regular

Um grafo é dito ser regular quando todos os seus vértices têm o mesmo grau.

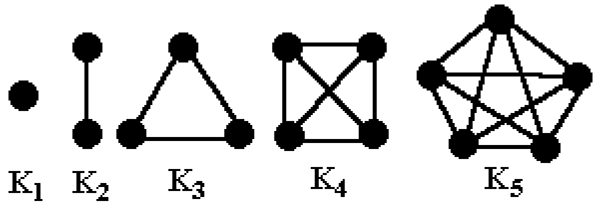
O grafo G4, por exemplo, é dito ser um grafo regular-3, pois todos os seus vértices têm grau 3.



G4

2.10 Grafo Completo

Um grafo é dito ser completo quando há uma aresta entre cada par de seus vértices. Estes grafos são designados por K_n, onde n é a ordem do grafo.



G4.1

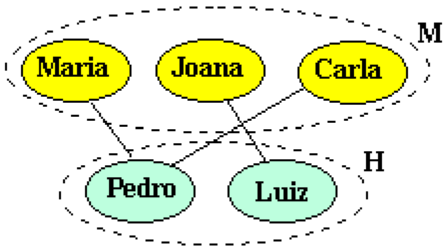
Um grafo K_n possui o número máximo possível de arestas para um dados n. Ele é, também regular-(n-1) pois todos os seus vértices têm grau n-1.

2.11 Grafo Bipartido

Um grafo é dito ser bipartido quando seu conjunto de vértices V puder ser particionado em dois subconjuntos V1 e V2, tais que toda aresta de G une um vértice de V1 a outro de V2.

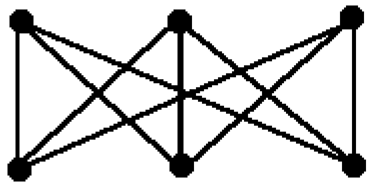
Para exemplificar, sejam os conjuntos H={h | h é um homem} e M={m | h é um mulher} e o grafo G(V,A) (ver o exemplo G5) onde:

V = H U M
A = {(v,w) | (v $\in$ H e w $\in$ M) ou (v $\in$ M e w $\in$ H) e <v foi namorado de w>}



G5

O grafo G6 é um K3,3, ou seja, um grafo bipartido completo que contém duas partições de 3 vértices cada. Ele é completo pois todos os vértices de uma partição estão ligados a todos os vértices da outra partição.



K3,3

G6

2.12 Grafo Rotulado

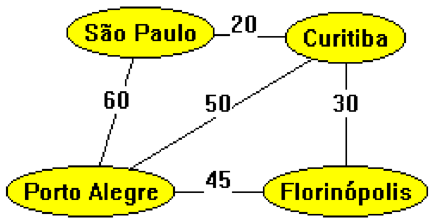
Um grafo G(V,A) é dito ser rotulado em vértices (ou arestas) quando a cada vértice (ou aresta) estiver associado um rótulo. G5 é um exemplo de grafo rotulado.

2.13 Grafo Valorado

Um grafo G(V,A) é dito ser valorado quando existe uma ou mais funções relacionando V e/ou A com um conjunto de números.

Para exemplificar (ver o grafo G7), seja G(V,A) onde:

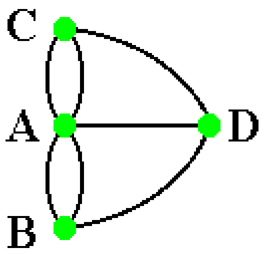
V = {v | v é uma cidade com aeroporto}
A = {(v,w,t) | <há linha aérea ligando v a w, sendo t o tempo esperado de vôo>}



G7

2.14 Multigrafo

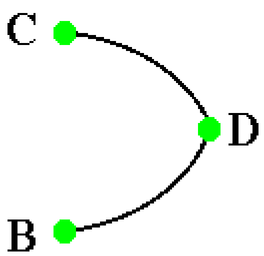
Um grafo G(V,A) é dito ser um multigrafo quando existem múltiplas arestas entre pares de vértices de G. No grafo G8, por exemplo, há duas arestas entre os vértices A e C e entre os vértices A e B, caracterizando-o como um multigrafo.



G8

2.15 Subgrafo

Um grafo Gs(Vs, As) é dito ser subgrafo de um grafo G(V,A) quando Vs⊂V e As⊂A. O grafo G9, por exemplo, é subgrafo de G8.



G9

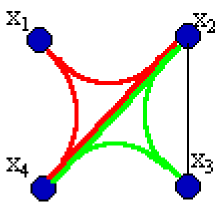
2.16 Hipergrafo

Um hipergrafo H(V,A) é definido pelo par de conjuntos V e A, onde:

V - conjunto não vazio;
A - uma família e partes não vazias de V.

Seja, por exemplo, o grafo H(V,A) dado por:

V = { x1, x2, x3, x4}
A = { {x1, x2, x4}, {x2, x3, x4}, {x2,x3}}



G10

2.17 Nomenclatura e Notação

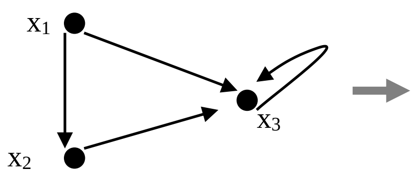
As formas de representação de Grafos apresentadas anteriormente são utilizadas para fins de apreciação visual ou por esquemas.

Para fins de cálculo, normalmente, associa-se ao grafo 3 tipos de matrizes.

a) Matriz de Adjacência - É uma matriz n X n, A = [a_{ij}], onde:

$a_{ij} = 1 \Leftrightarrow \exists\ (x_i, x_j)$
$a_{ij} = 0 \Leftrightarrow \pm\ (x_i, x_j)$

EXEMPLO

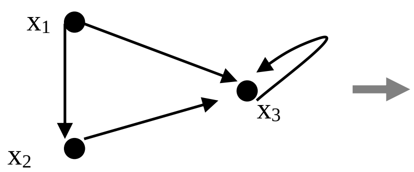


$$\begin{array}{c} \\ x1 \\ x2 \\ x3 \end{array} \begin{array}{c} \begin{array}{ccc} x1 & x2 & x3 \end{array} \\ \left| \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{array} \right| \end{array}$$

Esta é a matriz mais comumente utilizada.

b) Matriz Latina - é uma matriz figurativa onde os elementos são conjuntos de vértices. É usada em problemas de enumeração de caminhos. No exemplo acima, o mesmo grafo poderia ser reescrito como cadeias de caracteres que representem as associações entre os vértices.

EXEMPLO



$$\left| \begin{array}{ccc} & x_1x_2 & x_1x_3 \\ & & x_2x_3 \\ & & x_3x_3 \end{array} \right|$$

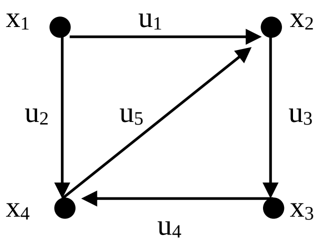
O trabalho computacional com matrizes latinas exige o uso de cadeias de caracteres, portanto, a linguagem a ser utilizada deveria fornecer facilidades neste tratamento. Por outro lado, ao contrário das matrizes de adjacência onde vértices sem arco eram representados por zero, em matrizes latinas são representadas por um caracter nulo.

c) Matriz de incidência - é uma matriz n x m, onde m é o número de arestas do grafo, onde cada coluna corresponde a um arco. A matriz de incidência é definida pela seguinte relação:

$b_{ij} = +1 \Leftrightarrow \exists\ (x_i, x_j) = u_j,\ x_i \neq x_k$
$b_{ij} = -1 \Leftrightarrow \exists\ (x_k, x_i) = u_j,\ x_k \neq x_i$
$b_{ij} = 0$ em todos outros casos

A relação definida na matriz de incidência apenas especifica se determinado vértice é extremidade inicial ou final de um arco u_j.

EXEMPLO



	u_1	u_2	u_3	u_4	u_5
x_1	+1	+1	0	0	0
x_2	-1	0	+1	0	-1
x_3	0	0	-1	+1	0
x_4	0	-1	0	-1	+1

Por utilizar mais espaço de memória (dada a maior quantidade de elementos a representar), a matriz de incidência é menos utilizada. No entanto, muito útil quando o problema a ser resolvido possui poucos vértices e arcos a considerar.

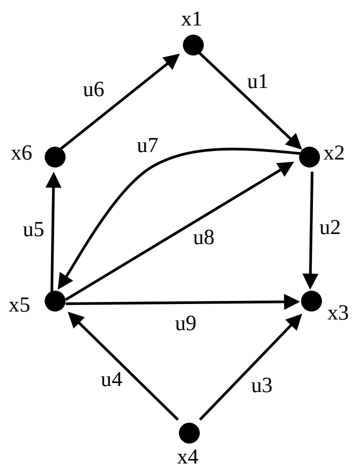
Exercícios

1) Desenhe grafos simples com 1,2, 3 e 4 vértices.

2) Construa a matriz de adjacência dos grafos do exercício 1.

3) Dado o grafo abaixo, construa:

a) matriz de adjacência
b) matriz latina
c) matriz de incidência



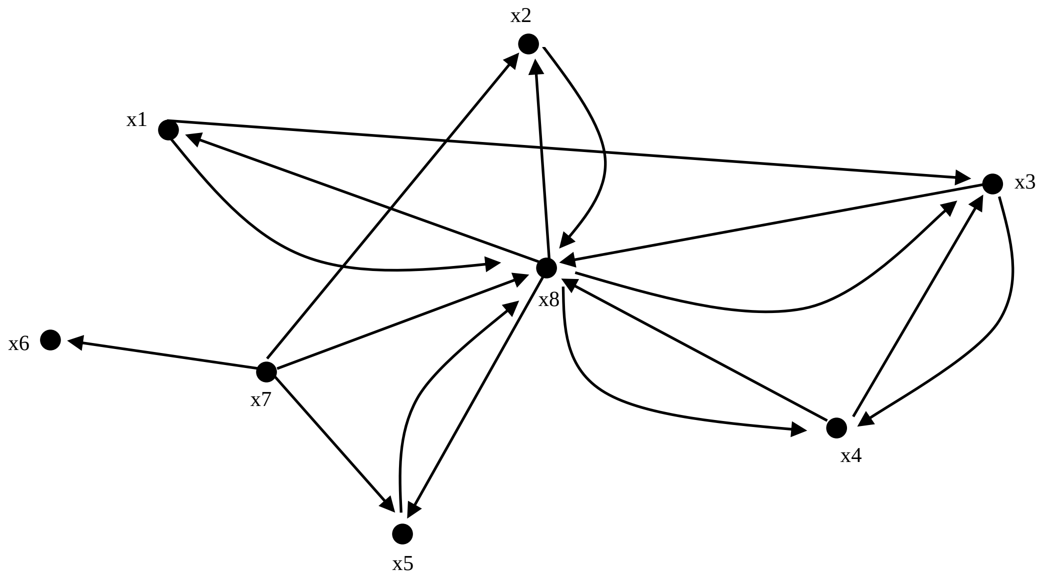
4) Dada da matriz de adjacência abaixo, construa o grafo que ela representa.

	a	b	c	d	e
a	0	1	0	1	0
b	0	0	0	1	1
c	0	1	0	0	1
d	0	0	1	0	0
e	1	0	1	0	0

5) Cinco turistas se encontram em um bar de Lages e começam a conversar, cada um falando de cada vez, com um só companheiro da mesa. O conhecimento de línguas dos turistas é mostrado na tabela a seguir.
Construa um grafo que represente todas as possibilidades de cada turista dirigir a palavra a outro, sendo compreendido. Defina que tipo de grafo foi obtido.

Turista	Inglês	Francês	Português	Alemão	Espanhol
1	X	X	X		X
2	X	X		X	
3		X	X	X	
4			X	X	X
5		X		X	X

6) O grafo abaixo representa as respostas colhidas em uma turma de crianças de escola na faixa de 7 anos, face à pergunta: "Quais são os colegas de quem você mais gosta?"



Expresse, usando a notação conveniente, os seguintes fenômenos observáveis no grafo:

a) posições de liderança
b) amizades recíprocas
c) criança com problemas de relacionamento
d) criança arredia

7) Faça um algoritmo para verificar a simetria de uma matriz. Ou seja, o algoritmo deve detectar em uma matriz A quando os elementos a(i,j) e a(j,i) são iguais.

2.18 Mais Alguns Conceitos

2.18.1 Cadeia

Uma cadeia é uma seqüência qualquer de arestas (ou arcos) adjacentes que ligam dois vértices. O conceito de cadeia vale também para grafos orientados, bastando que se ignore o sentido da orientação dos arcos. A seqüência de vértices (x6, x5, x4, x1) é um exemplo de cadeia em G11.

Uma cadeia é dita ser **elementar** se não passa duas vezes pelo mesmo vértice.

É dita ser **simples** se não passa duas vezes pela mesma aresta (arco).

O **comprimento** de uma cadeia é dado pelo número de arestas (arcos) que a compõe.

2.18.2 Caminho

Um caminho é uma cadeia na qual todos os arcos devem possuir a mesma orientação. Aplica-se, portanto, somente a grafos orientados. A seqüência de vértices (x1, x2, x5, x6, x3) é um exemplo de caminho em G11.

2.18.3 Ciclo

Um ciclo é uma cadeia simples e fechada (o vértice inicial é o mesmo que o vértice final). A seqüência de vértices (x1, x2, x3, x6, x5, x4, x1) é um exemplo de ciclo elementar em G11.

2.18.4 Circuito

Um circuito é um caminho simples e fechado. A seqüência de vértices (x1, x2, x5, x4, x1) é um exemplo de circuito elementar em G11.

2.18.5 Fecho Transitivo

Fechos transitivos indicam o inter-relacionamento direto ou indireto dos vértices.

Fecho Transitivo Direto (ftd) $(\hat{\Gamma}^{+}(x_i))$

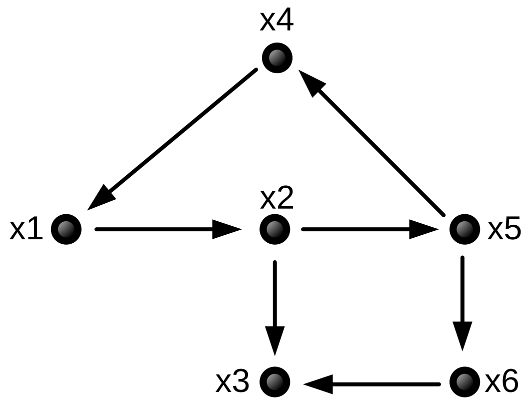
O fecho transitivo direto de um vértice v é o conjunto de todos os vértices que podem ser atingidos por algum caminho iniciando em v. O ftd do vértice x5 do grafo G12, por exemplo, é o conjunto: {x1, x2, x3, x5, x6}. Note que o próprio vértice faz parte do ftd já que ele é alcançável partindo-se dele mesmo.

Fecho Transitivo Inverso (fti) $(\hat{\Gamma}^{-}(x_i))$

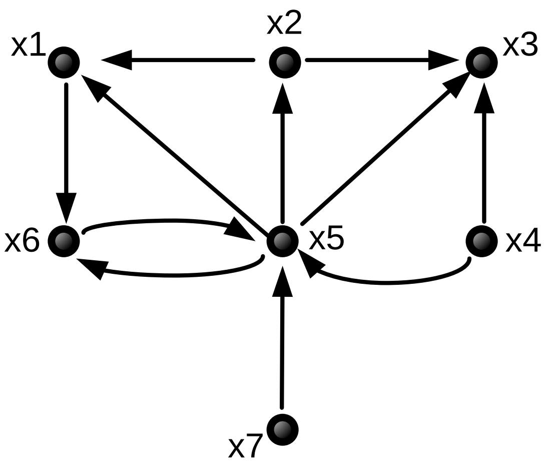
O fecho transitivo inverso de um vértice v é o conjunto de todos os vértices a partir dos quais se pode atingir v por algum caminho. O fti do vértice x5 do grafo G12, por exemplo, é o conjunto: {x1, x2, x4, x5, x7}. Note que o próprio vértice faz parte do fti já que dele se pode alcançar ele mesmo.

A noção de fechos transitivos está relacionada à provisão da capacidade de comunicação entre os vértices de um determinado grafo: até onde e a partir de onde podem estabelecer alguma conexão. (esta propriedade é muito útil em sistemas de comunicação)

Em ambos os casos (direto ou inverso), o próprio vértice faz parte do conjunto, pois, o mesmo pode ser atingido por ele mesmo em zero passos.



G11



G12

3 Conexidade e Distância

A noção de conexidade corresponde ao "estado da ligação" dos vértices de um grafo, ou seja, à possibilidade de se transitar de um vértice a outro. Dessa forma, podem ser estabelecidos critérios para a representação de problemas de acordo com o nível de agregação de seus componentes.

Problemas típicos solucionados com as técnicas que estudaremos neste capítulo são:
- determinação de localização de centros de emergência (ou seja, pontos ótimos para distribuição, tais como em armazéns, transportadoras, estabelecimentos comerciais e públicos, etc);
- problemas de engenharia de tráfego.

Convém salientar que as técnicas que estudaremos provêm duas estratégias:
- uma forma de modelar o problema (desenho de grafos);
- aplicação de teoremas e algoritmos para propor uma solução aos problemas modelados.

Dessa forma, pretende-se que as duas estratégias tenham sido compreendidas ao final deste capítulo.

3.1 Tipos de Conexidade

Quanto à conexidade, os grafos podem ser considerados conexos ou não conexos. A gradação da conexidade influencia no tipo de problema que se deseja modelar e resolver.

a) **Grafo Conexo** - Um grafo G(X,E) é dito ser conexo se há pelo menos uma cadeia ligando cada par de vértices deste grafo.

b) **Grafo Não Conexo (Desconexo)** - Um grafo G(X,E) é dito ser desconexo se há pelo menos um par de vértices que não está ligado por nenhuma cadeia.

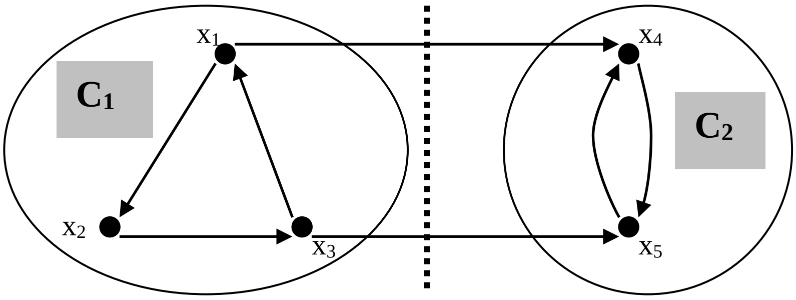
c) **Grafo Fortemente Conexo (f-conexo)** - No caso de grafos orientados, um grafo é dito ser fortemente conexo (f-conexo) se todo par de vértices está ligado por pelo menos um caminho em cada sentido, ou seja, se cada par de vértices participa de um circuito. Isto significa que cada vértice pode ser alcançável partindo-se de qualquer outro vértice do grafo. Como conseqüência, em um grafo f-conexo, tem-se sempre:

$$\hat{\Gamma}^{+}(x_i) = \hat{\Gamma}^{-}(x_i) = X, \forall x_i \in X$$

Grafos f-conexos merecem atenção especial, pois seu estudo permite identificar características importantes em modelos de problemas. Uma característica que é muita estudada é a das componentes f-conexas.

d) Componente Fortemente Conexa - Um grafo G(X,E) que não é fortemente conexo é formado por pelo menos dois subgrafos fortemente conexos.

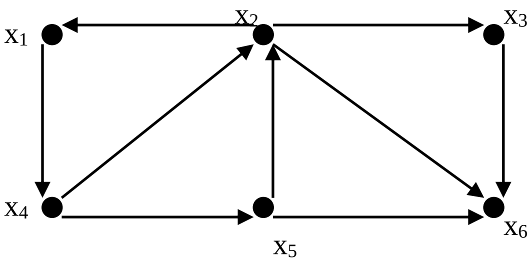
EXEMPLO



Acima se tem um grafo conexo com duas componentes f-conexas (C_1 e C_2). Se C_1 e C_2 fossem grafos separados seriam grafos f-conexos.

e) Grafo semi-fortemente conexo (sf-conexo) - é um grafo no qual para todo par de vértices x_i e x_j existe ao menos um dos dois caminhos (x_i, x_j) ou (x_j,x_i).

EXEMPLO



Em um grafo sf-conexo, a orientação, também não é importante. Trata-se de um caso generalizado, pois um grafo sf-conexo pode ser f-conexo, como simplesmente conexo.

3.2 Teorema de f-conexidade

O teorema abaixo é muito útil para elucidar características da f-conexidade. Algumas características nos induzem a um melhor entendimento do comportamento das componentes f-conexas.

Teorema: uma condição necessária e suficiente para que um grafo seja f-conexo é que todo subconjunto A de vértices de X, próprio e não-vazio, admita ao menos um sucessor que não pertença a A.

Dessa forma, pelo teorema temos:

$$\forall A \subset X, A \neq \varnothing: \quad \Gamma^{+}(A) - A \neq \varnothing$$

Demonstração:

($\Rightarrow$) sendo $A \subset X$ próprio e não vazio, podemos sempre escolher dois vértices a e b, tais que a $\in A$ e $b \notin A$. Se A não admitir descendentes fora de A, não existirá um caminho entre a e b e o grafo não será f-conexo

($\Leftarrow$) É preciso demonstrar que dado um $x_i \in X$ qualquer, os dois conjuntos $\hat{\Gamma}^+(x_i)$ $\hat{\Gamma}^-(x_i)$

se identificam com X ao se considerar válida a condição do teorema.
Esta prova é feita por absurdo (BOAVENTURA NETTO, 1979, pp. 38)

3.3 Grafo Reduzido

Dado um grafo G = (X,E). Gr = (S,W) é um grafo reduzido de G segundo uma partição de S se, após uma ordenação conveniente, tem-se:

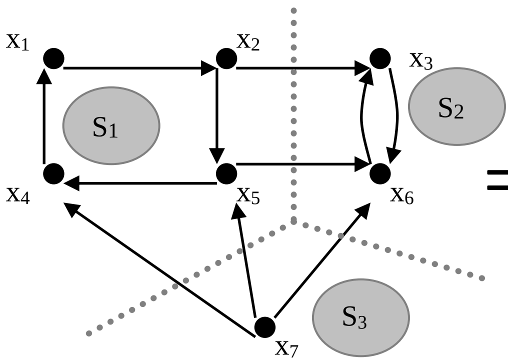
$S_i = \{ x_k \mid x_k \in X\}$, $S_i \cap S_j = \varnothing \;\; \rightarrow \bigcup_i S_i = X$

$W = \{ (S_i, S_j) \mid \exists\, (x_k, x_l) \in U$, sendo que $x_k \in S_i$ e $x_l \in S_j\}$

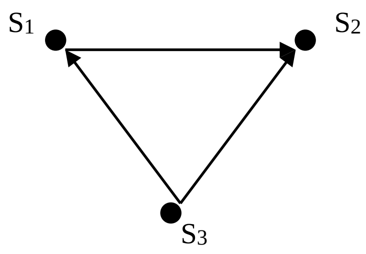
A partição a ser feita depende da relação de equivalência considerada; ao se reduzir a escala de um mapa, por exemplo, pode ser conveniente a agregação de uma cidade maior e suas satélites, a fim de não prejudicar a clareza (partição por proximidade geográfica).
A partição mais comumente estudada com auxílio da técnica da redução é a baseada nas componentes f-conexas.

O grafo reduzido será, então, construído agregando-se em um só vértice todos os vértices de cada componente f-conexa e, substituindo-se todos os arcos que unirem cada par de componentes por um único arco.

EXEMPLO



$\Rightarrow$



É perceptível que existem 3 componentes f-conexas no grafo acima. O propósito é reduzir as componentes f-conexas a um único vértice, representados no grafo reduzido ao lado.

Cada componente (S_1, S_2 e S_3) representa um conjunto de vértices, mais especificamente, os vértices de uma componente f-conexa.

Observação:

- A substituição dos arcos entre duas componentes f-conexas por um único é sempre possível visto que todos eles devem ter, OBRIGATORIAMENTE, o mesmo sentido, ou então haveria um caminho de ida e volta, portanto, teríamos apenas uma componente e não duas.

3.4 Critérios de Conexidade

Assim como o teorema enunciado e demonstrado na seção 3.2, os critérios enunciados aqui, na forma de um teorema, serão utilizados como verdades na construção de algoritmos relacionados ao tratamento da conexidade.

Critério da Conexidade Forte
Teorema: as três proposições abaixo são equivalentes para todo grafo G = (X,E) não trivial.
(ou seja, a $\Rightarrow$ b, b $\Rightarrow$ c e c $\Rightarrow$ a).

(a) G é f-conexo;
(b) G possui um caminho fechado passando por todos os vértices (circuito pré-hamiltoniano);
(c) Gr (Grafo Reduzido) tem apenas um vértice.

3.5 Algoritmos para decomposição de um grafo em componentes f-conexas

Há diversos recursos disponíveis para se saber se um grafo é ou não f-conexo e para decompô-lo em componentes f-conexas, caso ele não seja f-conexo. Em muitos casos, a determinação de algumas propriedades pode fornecer essa informação como subproduto, mas, se apenas se deseja a decomposição, é conveniente usar um algoritmo específico.

3.5.1 Algoritmo de Malgrange

Este algoritmo é baseado na determinação de fechos transitivos. Onde a interseção de fechos transitivos (direto ou inverso), detecta a presença de uma componente f-conexa.

De forma generalizada, o algoritmo possui os seguintes passos:

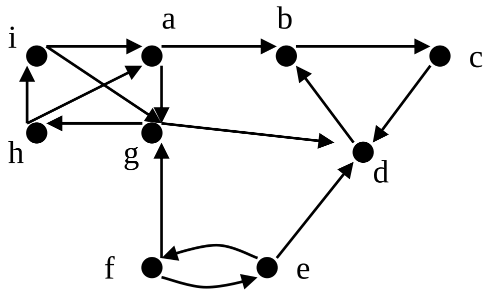
[1] determinar os fechos transitivos de um vértice x_i e efetuar a sua interseção;
[2] eliminar da matriz de adjacência as linhas e colunas correspondentes aos vértices obtidos na interseção no passo 1;
[3] o processo recomeça com a escolha de outro vértice x_j até que a matriz tenha se esgotado.

Observações:

- se a matriz se esgotar na primeira iteração, o grafo será f-conexo;
- o vértice escolhido sempre será eliminado na matriz de adjacência, haja visto que o mesmo participa de ambos os fechos transitivos e, portanto, participará da interseção;
- quando se tratar de uma matriz de adjacência muito grande, o tempo de computação será elevado.

EXEMPLO

O exemplo abaixo, baseia-se na matriz de adjacência formada a partir do grafo e possui 3 iterações (mostradas abaixo) até que sejam detectadas todas as componentes f-conexas.



	a	b	c	d	e	f	g	h	i
a		1					1		
b			1						
c				1					
d		1							
e				1		1			
f					1		1		
g				1				1	
h	1								1
i	1						1		

Iteração [1]

Vértice escolhido: a

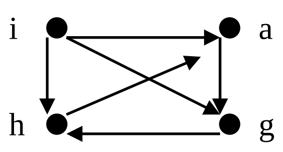
$\hat{\Gamma}^{+}(a) = \{a,b,c,d,g,h,i\}$

$\hat{\Gamma}^{-}(a) = \{a,e,f,g,h,i\}$

$\hat{\Gamma}^{+}(a) \cap \hat{\Gamma}^{-}(a) = \{a,g,h,i\}$

	b	c	d	e	f
b		1			
c			1		
d	1				
e			1		1
f				1	

Dessa forma, a primeira componente f-conexa identificada é a formada pelos vértices {a,g,h,i}, mostrada no grafo ao lado. E a matriz de adjacência resultante não possuirá tais vértices.



Iteração [2]

Vértice escolhido: b

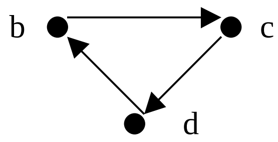
$\hat{\Gamma}^{+}(b) = \{b,c,d\}$

$\hat{\Gamma}^{-}(b) = \{b,c,d,e,f\}$

$\hat{\Gamma}^{+}(b) \cap \hat{\Gamma}^{-}(b) = \{b,c,d\}$

	e	f
e		1
f	1	

Dessa forma, a segunda componente f-conexa identificada é a formada pelos vértices {b,c,d}, mostrada no grafo ao lado. E a matriz de adjacência resultante não possuirá tais vértices.



Iteração [3]

Vértice escolhido: e

$\hat{\Gamma}^{+}(e) = \{e,f\}$

$\hat{\Gamma}^{-}(b) = \{e,f\}$

$\hat{\Gamma}^{+}(b) \cap \hat{\Gamma}^{-}(b) = \{e,f\}$

Dessa forma, a terceira componente f-conexa identificada é a formada pelos vértices {e,f}, mostrada no grafo ao lado. E a matriz de adjacência resultante tornar-se-á vazia.

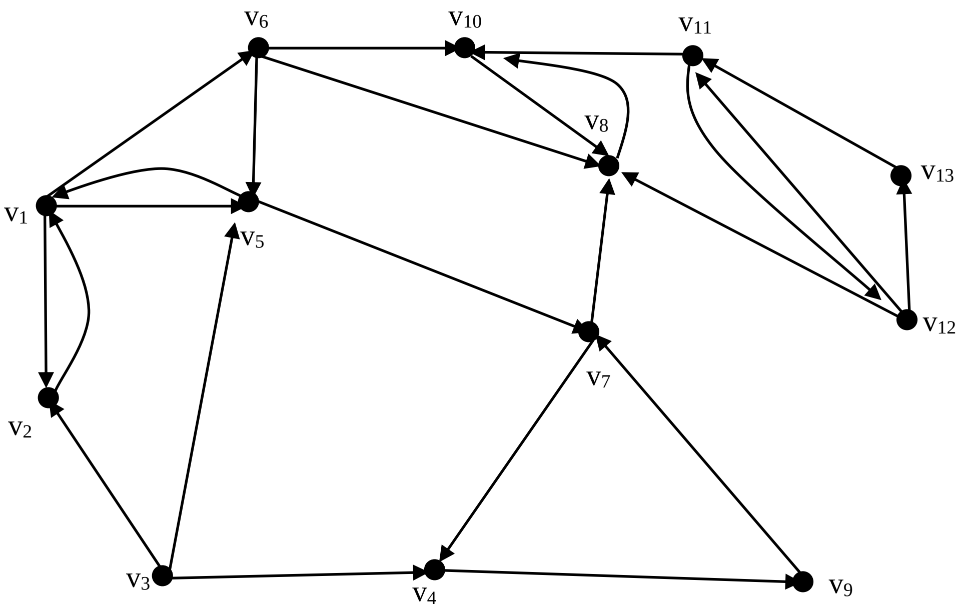
Observações:
As contribuições obtidas do algoritmo de Malgrange são:

- identificação do tipo de conexidade do grafo (conexo, porém com três componentes f-conexas);
- grafo reduzido do grafo original, formado pelas 3 componentes identificadas acima.

NOTA: Consideremos que G' é uma componente fortemente conexa, algum de seus vértices podem aparecer em outra componente G" também fortemente conexa? Obviamente, NÃO!! Pois se v_i aparece em duas (ou mais) componentes fortemente conexas G' e G", então existe um caminho de um vértice $v_i \in G'$ a outro vértice $v_j \in G''$ e vice-versa. Portanto, a união das duas componentes será fortemente conexa, o que é uma contradição em relação à definição.

Exercícios

1) Encontre as componentes fortemente conexa e o grafo reduzido do seguinte dígrafo.



2) Determine o tipo de conexidade dos seguintes grafos e cassifique-os observando a sua forma:

a) V = {cidades dos Brasil}
   A = { (x_i, x_j) / <x_i fala por DDD com x_j>}

b) V = {livros de uma biblioteca}
   A = { (x_i, x_j) / <x_i é do mesmo autor que x_j>}

3.5.2 Outra forma de encontrar as componentes conexas

As componentes fortemente conexas de um grafo, podem ser determinadas também, diretamente, pelo uso das matrizes R e Q.

A matriz R= [r_{ij}] é definida como:

$$r_{ij} = \begin{Bmatrix} 1, \text{se o vértice } v_j \text{ } \textit{pode} \text{ ser atingido partindo - se do vértice } v_i; \\ 0, \text{caso contrário} \end{Bmatrix}$$

E a matriz Q é definida como a matriz transposta de R, isto é, $Q = R^T$. Faz-se $R \otimes Q$ para achar as componentes fortemente conexas. O produto é feito elemento a elemento (linha por linha).

3.6 Base e Anti-base

Definição 1
[**Boaventura**]

Uma base de um grafo $G = (X,\Gamma)$ é um subconjunto B de vértices, tais que:

a) dois vértices quaisquer de B não são ligados por nenhum caminho;
b) todo vértice não pertencente a B pode ser atingido por um caminho partindo de B.

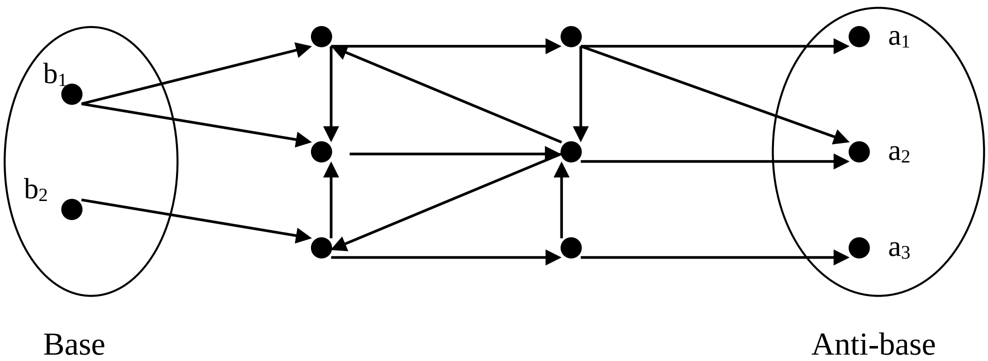
Definição 2
[**Rabuske**]

Uma base B de um dígrafo é um subconjunto de vértices que não podem ser alcançados a partir de outros vértices deste dígrafo. Este conjunto é MINIMAL no sentido de que nenhum subconjunto de B tem esta propriedade.

Uma anti-base de um grafo é um subconjunto A de vértices possuindo a propriedade (a) da definição 1 e tendo a propriedade (b) invertida, ou seja:

de todo vértice não pertencente a A se pode atingir algum vértice em A através de um caminho.

EXEMPLO



Teorema: Todo grafo possui uma base; se $C_1, ..., C_r$ são componentes f-conexas correspondendo a vértices **sem antecessores** no grafo reduzido de um grafo G, e se $C_i = (X_i, U_i)$, então as bases de G correspondem aos elementos do produto cartesiano $X_i \times ... \times X_r$.

Demonstração:

- como todo vértice não pertencente a B pode ser atingido a partir de B, a base é maximal em relação à propriedade (a), visto que todo vértice não atingível deve pertencer a B;

- por outro lado, como não existem caminhos entre os vértices de B, a base é minimal em relação a (b) a qual se verifica para todo subconjunto de X que contenha B;

- toda base deve conter, portanto, um e um só vértice de cada componente f-conexa sem antecessores e qualquer base será, portanto, da forma $\{x_a, x_b, ..., x_k\}$ onde $x_a \in X_1$, $x_b \in X_2$, .. $x_k \in X_r$, ou seja, será um elemento do produto cartesiano das componentes conexas;

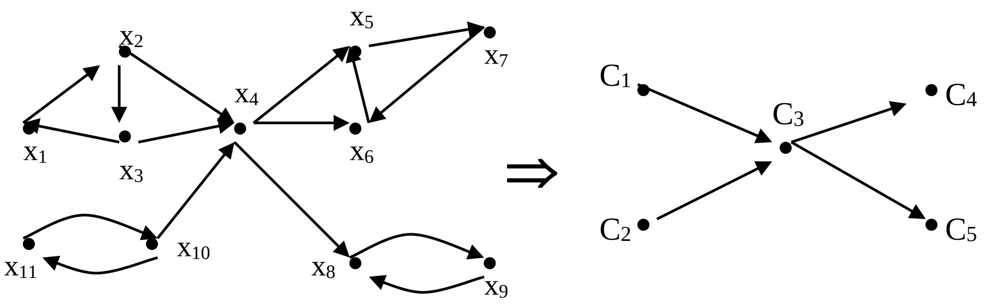
- como existe ao menos uma componente conexa para todo grafo, pode-se afirmar que todo grafo possui uma base.

Pode-se enunciar um teorema semelhante para as anti-bases, substituindo-se "antecessores" por sucessores.

Observações:

- Se o grafo for f-conexo, cada componente do grafo isoladamente, é considerado uma base;
- A noção de base e anti-base, permite a visualização da distribuição de serviços, num problema de transporte, por exemplo, modelado como um grafo.

EXEMPLO



As componentes f-conexas são facilmente identificadas, no entanto, de acordo com o teorema, apenas nos interessa as componentes que não possuam antecessores. Dessa forma, a base a ser formada deveria possuir elementos apenas das componentes C_1 e C_2.

Pode-se observar, através do exemplo acima, que a ligação por meio de novos arcos, de um número suficiente de pares bases/anti-bases tornará o grafo f-conexo. Dado o interesse da ocorrência da f-conexidade em diversos problemas (por exemplo, de engenharia de trânsito), é interessante que se discuta a possibilidade de tornar o grafo f-conexo através da adição de um número mínimo de arcos ao grafo original.

A restauração da f-conexidade após a adoção da mão única em parte de um conjunto de ruas, ou em todo o conjunto, seria um problema onde a minimização do número de arcos teria grande importância, dado o custo geralmente elevado dessa implantação (por se tratar, geralmente de ruas que deverão ser criadas).

Aplicando-se os dois teoremas seguintes, pode-se construir um algoritmo com esta finalidade:

3.6.1 Teorema 1

Seja G = (X, U) um grafo não f-conexo e sejam A e B respectivamente uma anti-base e uma base de G. Dados dois vértices a $\in$A e b$\in$ B, vamos definir:

A' = A - {a}, se A $\neq$ {a}
A se A = {a}

B' = B - {b}, se B $\neq$ {b}
B se B = {b}

Então, A' e B' serão respectivamente uma anti-base e uma base de G' = (X,U') onde U' = U $\cup$ {(a,b)} se e somente se:

- a possuir ao menos um ascendente em B';
- b possuir ao menos um descendente em A'.

Dessa forma, o teorema fornece um critério para a adição iterativa de arcos, atendendo ao requisito (a).

3.6.2 Teorema 2

O número mínimo de arcos a serem adicionados a um grafo G, não f-conexo, para torná-lo f-conexo, é igual a:

t = max[a(G), b(G)], onde a(G) e b(G) são respectivamente, os cardinais das anti-bases e bases de G.

3.7 Algoritmo

O algoritmo resultante da aplicação dos teoremas (apresentados nas seções 3.6.1 e 3.6.2) são utilizados para que um grafo não f-conexo seja transformado em um grafo f-conexo, conforme os seguintes passos:

[1] Seja G_0 o grafo inicial;
[2] Determinam-se uma anti-base A_0 e uma base B_0 de G_0;
[3] Constrói-se uma matriz de a(G_0) linhas e b(G_0) colunas, denominada C_0;
[4] Faz-se corresponder à linha i, o elemento a_i de A_0 e à coluna j o elemento b_j de B_0.

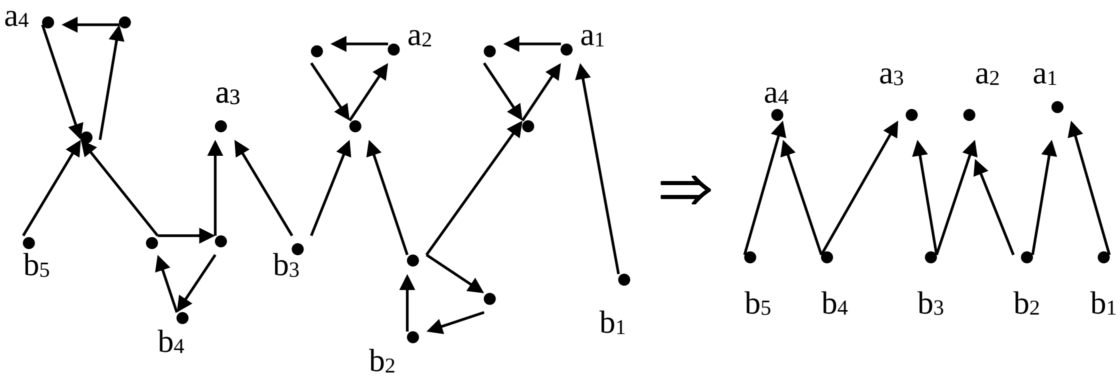
De acordo com o passo [4], os valores dos elementos da matriz C_0 serão dados por:

$c_{ij} = 1$ se b_j é ascendente de a_i;
$c_{ij} = 0$ caso contrário

O exemplo abaixo identifica a montagem da matriz C_0 e os outros passos que devem ser seguidos para que o algoritmo produza o grafo f-conexo do grafo original.

Para que a matriz C_0 seja criada, antes devemos identificar uma base e uma anti-base. De acordo com o teorema esboçado, bases e anti-bases podem ser inferidas a partir da determinação de componentes f-conexas e, conseqüentemente, da análise de seu grafo reduzido. Este passo está sendo executado no exemplo abaixo:

EXEMPLO



De acordo com a redução efetuada, os vértices identificados por b_1, ..., b_5 podem compor bases, assim como os vértices identificados por a_1, ..., a_4 podem compor anti-bases. Dessa forma, a matriz C_0 criada é:

$C_0 =$

	b_1	b_2	b_3	b_4	b_5
a_1	1	1			
a_2		1	1		
a_3			1	1	
a_4				1	1

Para tornar o grafo f-conexo, duas questões surgem:

- Quantos arcos, no mínimo, são necessários para que o grafo se torne f-conexo?
  A resposta a esta questão está no Teorema 3.6.2. $t = \max[a(G), b(G)]$, ou seja, o número t (mínimo de arcos a serem adicionados) é: $\max[4,5] = 5$.

- Como alocar os novos cinco arcos?
  A resposta é obtida através das duas regras que são discutidas abaixo.

Regra [1] A regra 1 deve ser aplicada iterativamente, até que a matriz se esgote.

i. escolha um arco nulo $(a_i,b_j) = 0$ ou $c_{ij} = 0$;
ii. analise todos os elementos da linha a_i e da coluna b_j, procurando por aqueles cujo valor é 1. Suprimindo a linha i e a coluna j de C_r após colocar 1 em toda casa nula C_{hk}^r tal que $C_{hj}^r = C_{ik}^r = 1$.

Regra [2] Se Cr é a última matriz construída e se não existe nenhuma posição nula, Cr terá m_r linhas e n_r colunas. Constroem-se, então, pares (a_i,b_j) em número igual a max (m_r,n_r) tais que cada a_i e cada b_j participem de, ao menos, um par.

Aplicando as regras 1 e 2 ao exemplo indicado, terão 4 iterações.

Iteração [1]

	b_1	b_2	b_3	b_4	b_5
a_1	1	1			
a_2	*	1	1		
a_3			1	1	
a_4				1	1

Porque escolhemos $c_{hk}= c_{13} = (b_3,a_1)$?

Porque de acordo com a regra 1, necessitaríamos um c_{hk}, tal que existe c_{hj} e c_{ik}.

se chk = c13, então, h=1 e k=3.

Temos c22=c23=c11=1. Onde teríamos, c1j = ci3 = 1?

c11=c23 apenas.

Logo c13 é um caminho viável e pode ser construído.

Aplicando a Regra 1 ...

Se escolhermos o arco (a_2,b_1), ou seja, c_{21}, ao analisarmos a figura do grafo, veremos que o mesmo não existe, portanto, $(a_2,b_1) = 0$.
Marcamos na matriz com um asterisco.
Devemos então analisar os elementos c_{h1} (coluna de b_1) e c_{2k} (linha de a_2).

Coluna de b_1: $c_{11} = 1$ $(b_1,a_1) = 1$
Linha de a_2: $c_{22} = c_{23} = 1$ $(b_2,a_2)=(b_3,a_2)=1$

Possível c_{hk}: $c_{13} = 0$ $(b_3,a_1) = 0$

A inclusão de (a_2,b_1) gera um caminho (b_3,a_1) ? SIM.

Então, para a próxima iteração:
Na matriz Co: $c_{13} = 1$ e excluir linha a_2 de e coluna de b_1.
No grafo original incluir o arco (a_2,b_1).

Iteração [2]

	b_2	b_3	b_4	b_5
a_1	1	1*		
a_3	*	1	1	
a_4			1	1

* marcado na iteração anterior

Porque escolhemos $c_{hk}= c_{14} = (b_4,a_1)$?

De acordo com a análise da regra 1, necessitaríamos um chk, tal que existe chj e cik.

se chk = c14, então, h=1 e k=4.
Temos c33=c34=c12=1. Onde teríamos, c1j = ci4 = 1?

c12=c34 apenas.

Logo c14 é um caminho viável e pode ser construído.

Aplicando a Regra 1 ...

Se escolhermos o arco (a_3,b_2), e analisarmos a figura do grafo, veremos que o mesmo não existe, portanto, $(a_3,b_2) = 0$.

Coluna de b_2: $c_{12} = 1$ $(b_2,a_1) = 1$
Linha de a_3: $c_{33} = c_{34} = 1$ $(b_3,a_3)=(b_4,a_3)=1$

Possíveis c_{hk}: $c_{14} = 0$ $(b_4,a_1) = 0$
A inclusão de (a_3,b_2) gera um caminho de (b_4,a_1)? Sim

Então, para a próxima iteração:
Na matriz Co: $c_{14} = 1$ e excluir linha de a_3 e a coluna de b_2.
No grafo original incluir o arco (a_3,b_2).

Iteração [3]

	b_3	b_4	b_5
a_1	1	1*	
a_4	*	1	1

* marcado na iteração anterior

```
Porque escolhemos chk= c15 = (b5,a1) ?

De acordo com a análise da regra 1, necessitaríamos um chk, tal que existe chj e cik.

se chk = c15, então, h=1 e k=5.
Temos c44=c45=c13=1. Onde teríamos, c1j = ci5 = 1?

c13=c45 apenas.

Logo c15 é um caminho viável e pode ser construído.
```

Aplicando a Regra 1 ...

Se escolhermos o arco (a_4,b_3), e analisarmos a figura do grafo, veremos que o mesmo não existe, portanto, $(a_4,b_3) = 0$.

Coluna de b_3: $c_{13} = 1$ $(b_3,a_1) = 1$
Linha de a_4: $c_{44} = c_{45} = 1$ $(b_4,a_4)=(b_5,a_4)=1$

Possíveis c_{hk}: $c_{15} = 0$ $(b_5,a_1) = 0$ e $c_{14} = 1$ $(b_4,a_1) = 1$ (da iteração anterior), portanto, analisaremos c_{15}.

A inclusão de (a_4,b_3) gera um caminho (b_5,a_1) ? SIM.

Então, para a próxima iteração:
Na matriz Co: $c_{15} = 1$ e excluir linha a_4 de e coluna de b_3.
No grafo original incluir o arco (a_4,b_3).

Iteração [4]

	b_4	b_5
a_1	1	1*

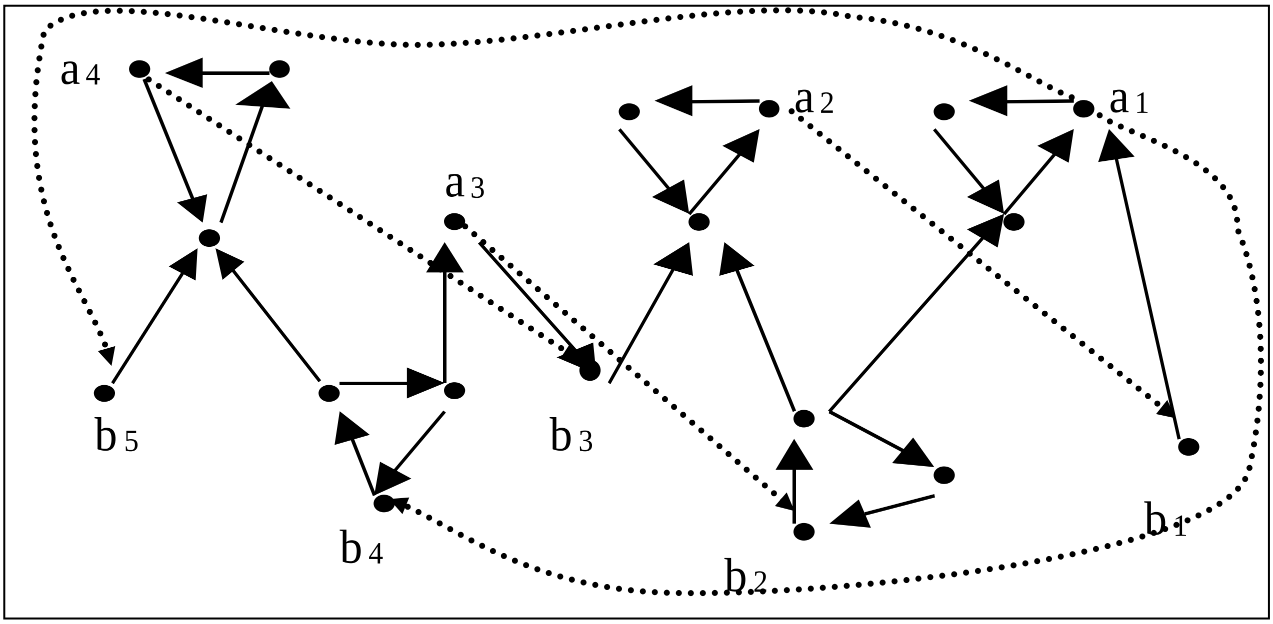
* marcado na iteração anterior

De acordo com a Regra 2, devem ser construídos mais max[m_r, n_r] arcos. Da matriz acima verifica-se que max[1,2] = 2, portanto, teremos mais dois arcos.

Aplicando a Regra 2 ...

A matriz não suporta mais a construção de caminhos intermediários, portanto, serão incluídos no grafo original os arcos: (a_1,b_4) e (a_1,b_5).

Dessa forma, o grafo f-conexo produzido será o seguinte:



4. Distância

Dados dois vértices v e w pertencentes ao grafo G(V,A) denomina-se distância, entre v e w, ao comprimento do menor caminho entre esses dois vértices. No caso da não existência desse caminho, considera-se a distância infinita.

Será usado d(v,w) como notação de distância entre os vértices v e w.

4.1 Excentricidade (afastamento)

Denotado por e(v), de um vértice v, é a máxima das distâncias d(v,u), isto é, $\forall$ u $\in$ G, e(v) = max d(v,u).

4.2 Raio

Denotado por r(G) de um grafo G é o min e(v).

4.3 Centro

De um grafo G é definido pelo conjunto de vértices v tais que e(v) = r(G).

4.4 Localização do Centro de Emergência

O interesse é em determinar o centro de um grafo de tal modo que o tempo de ida e volta seja mínimo (é o caso da localização de hospitais, polícia, bombeiros, etc).

A excentricidade e o raio serão respectivamente determinados como:

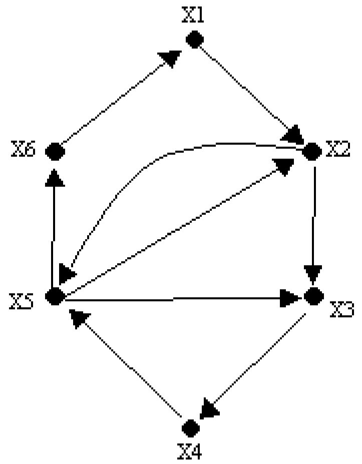
$e_{sr}(x_i) = \max \{v_j[d(x_i, x_j) + d(x_j, x_i)]\}\ x_i \in V$

$r(G) = \min [e_{sr}(x_i)]\ x_i \in V$

Onde:

O índice sr significa saída e retorno, v_j é o peso do vértice x_i.

EXEMPLO



Determinar o Centro de Emergência do Grafo ao lado.

D(G)=

	x1	x2	x3	x4	x5	x6	$e_s(x_i)$
x1	0	1	2	3	2	3	3
x2	3	0	1	2	1	2	3
x3	4	3	0	1	2	3	4
x4	3	2	2	0	1	2	3
x5	2	1	1	2	0	1	2 *
x6	1	2	3	4	3	0	4
$e_r(x_i)$	4	3 *	3 *	4	3 *	3 *	

Logo, o centro de emergência é determinado através de $D(G) + D(G)^T$

$D(G) + D(G)^T$ =

	x1	x2	x3	x4	x5	x6	$e_{sr}(x_i)$
x1	0	4	6	6	4	4	6
x2	4	0	4	4	2	4	4
x3	6	4	0	3	3	6	6
x4	6	4	3	0	3	6	6
x5	4	2	2	3	0	4	4
x6	4	4	6	6	4	0	6

Logo, o raio de G é 4 e o centro de emergência de G é formado pelo conjunto {x2, x5}, pois pela definição de centro, temos que o centro de G é definido pelo conjunto de vértices v tais que e(v) = r(G).

4.5 Teorema Sobre Matrizes Booleanas de adjacência e Alcançabilidade

Se A é a matriz booleana de adjacências para um grafo direcionado G com e vértices e sem arcos paralelos, então:
$A^{[m]}[i,j] = 1$ se e somente se houver um caminho de comprimento m do vértice vi ao vértice vj.

Portanto, basta efetuar a multiplicação matricial da matriz de adjacência para verificar os caminhos existentes no grafo.

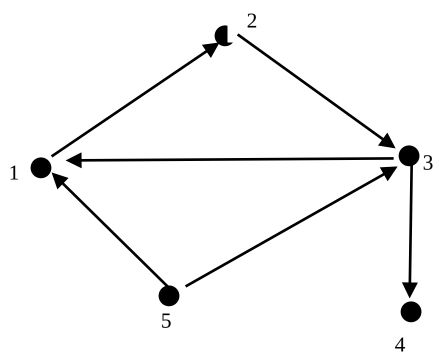
OBS.: Se houver n vértices no grafo, então qualquer caminho com n+1 ou mais vértices deve ter pelo menos um vértice repetido. Conseqüentemente, nunca devemos procurar caminhos com comprimentos maior que n.

Então, pode-se definir a matriz de alcançabilidade R como:

$R= A^{(1)} \vee A^{(2)} \vee \ldots \vee A^{(n)}$

Onde: n é o número de vértices, ou seja, vj é alcançável a partir de vi, se e somente se, o elemento i, j em R for igual a 1.

EXEMPLO



Construir a matriz de alcançabilidade R, do grafo abaixo.

5. Caminhos

Neste capítulo estudar-se-á os seguinte itens:

- Caminhos Hamiltonianos (Caixeiro Viajante);
- Problemas Eulerianos (Carteiro Chinês);
- Menor Caminho entre dois vértices;

Vamos relembrar o que é Cadeia?

Cadeia é qualquer seqüência de arestas onde o vértice final de uma aresta é o vértice inicial da próxima.

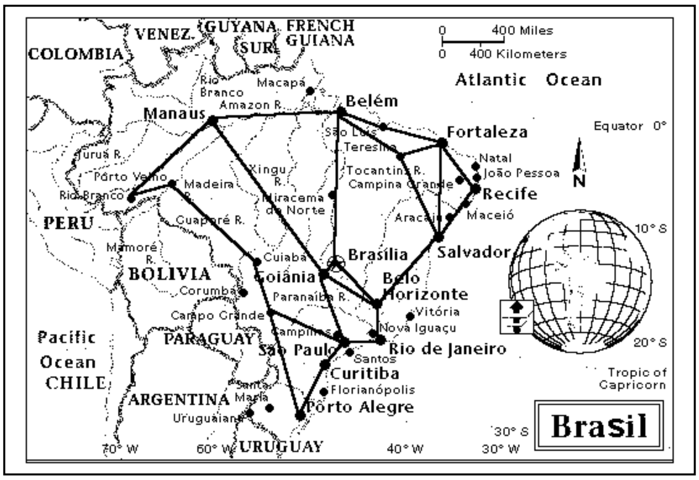
Como uma cadeia de k vértices é formado por (k-1) arestas, diz-se que o valor (k-1) é o comprimento da cadeia. Ou seja, conforme visto anteriormente, o tamanho de uma cadeia é dado pelo número de arestas percorridas.

Relembrando o conceito de caminho: Um caminho é uma cadeia na qual todos os arcos possuem a mesma orientação.

Podemos deduzir que em grafos conexos não orientados e ponderados, sempre existe um caminho entre quaisquer dois vértices x e y. De fato, pode haver vários desses caminhos. A pergunta é como encontrar um caminho com o menor peso? Como os pesos geralmente representam distâncias, este problema ficou conhecido como problema do "caminho mínimo".

5.1 Problema do Caminho de Custo Mínimo

De forma a reduzir seus custos operacionais, uma empresa de transporte de cargas deseja oferecer aos motoristas de sua frota um mecanismo que os auxilie a selecionar o melhor caminho (o de menor distância) entre quaisquer duas cidades por ela servidas, de forma a que sejam minimizados os custos de transporte.



Um possível modelo para este problema poderia ser:

G (V,A)
V = { x | x é cidade }
A = { (xi, xj, d) | há uma conexão por estrada entre as cidades xi e xj, sendo d a distância que as separa }

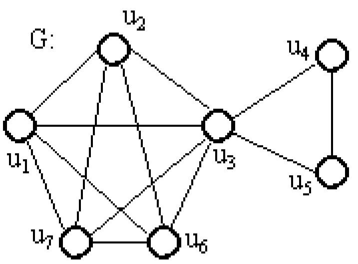
O problema de encontrar o caminho de custo mínimo entre dois vértices de um grafo é o mais importante relacionado com a busca de caminhos em grafos em vista de sua aplicação a várias situações da realidade.

Assim sendo, é importante termos soluções computacionais viáveis para resolver problemas desse tipo.

Há um grande número de situações possível quando da obtenção deste caminho, a exemplo de: existência ou não de ciclos; determinação do caminho ou apenas do custo mínimo. Dada esta diversidade de situações, há um número razoável de algoritmos que foram propostos ao longo do tempo, dentre os quais os algoritmos de Dijkstra e de Floyd se destacam.

5.2 Grafos Eulerianos

Um grafo G é dito ser euleriano se há um ciclo em G que contenha todas as suas arestas. Este ciclo é dito ser um ciclo euleriano. O grafo da figura abaixo, por exemplo, é euleriano já que ela contém o ciclo: (u1, u2, u3, u4, u5, u3, u1, u6, u2, u7, u3, u6, u7, u1), que é euleriano.



O teorema abaixo provê uma solução simples para determinar se um grafo é euleriano:

Teorema: Existe um caminho euleriano em um grafo conexo se, e somente se, não houver nenhum ou existirem exatamente dois vértices de grau ímpar. No caso de não haver vértices de grau ímpar, o caminho pode começar em qualquer vértice e terminará neste mesmo vértice; para o caso de haver dois vértices de grau ímpar, o caminho deve começar em um vértice ímpar e terminar no outro.

O teorema dos caminhos de Euler é, na verdade, um algoritmo para determinação se um caminho de Euler existe ou não em um grafo conexo arbitrário.

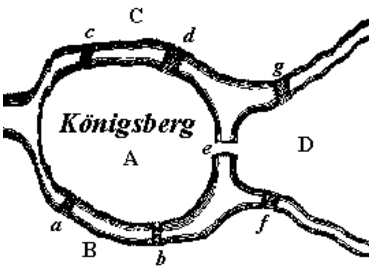
Exercício 1: Escreva um algoritmo em pseudocódigo para determinar se um grafo é de Euler ou não.

Provavelmente, o exemplo mais antigo de problema que faz uso de grafos (ou conceito correlatos) como modelo matemático é o Problema das Pontes de Königsberg. Ele foi resolvido pelo matemático suiço Leonhard Euler (1707-1783) em 1736.

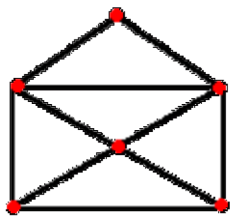
5.2.1 Problema das Pontes de Königsberg

No século 18 havia na cidade de Königsberg um conjunto de sete pontes (identificadas pelas letras de a até f na figura abaixo) que cruzavam o rio Pregel . Elas conectavam duas ilhas (A e D) entre si e as ilhas com as margens (B e C).

Por muito tempo os habitantes daquela cidade perguntavam-se se era possível cruzar as sete pontes numa caminhada contínua sem que se passasse duas vezes por qualquer uma delas.



5.2.2 Problema do desenho da casa

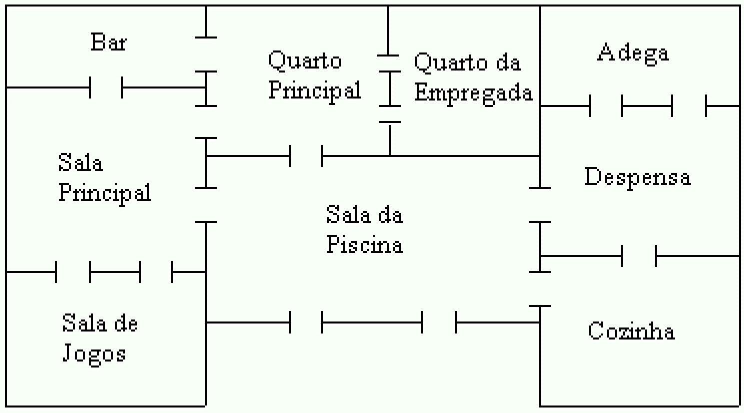


No desenho acima, uma criança diz ter posto a ponta do lápis numa das bolinhas e com movimentos contínuos (sem levantar e sem retroceder o lápis) traçou as linhas que formam o desenho da casa, traçando cada linha uma única vez.

Exercício 2: Nos dois exemplos anteriores, determine se existe ou não um caminho de Euler.

5.3 Grafos Hamiltonianos

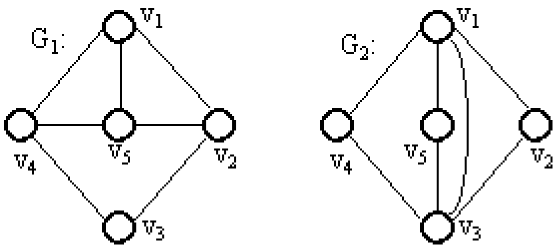
5.3.1 O Caso Count Van Diamond



O cenário acima é a residência do bilionário Count Van Diamond, que acaba de ser assassinado. Sherlock Gomes (um conhecido detetive que nas horas vagas é um estudioso da Teoria dos Grafos) foi chamado para investigar o caso. O mordomo alega ter visto o jardineiro entrar na sala da piscina (lugar onde ocorreu o assassinato) e logo em seguida observou que ele saiu daquela sala pela mesma porta que havia entrado. O jardineiro, contudo, afirma que ele não poderia ser a pessoa vista pelo mordomo, pois ele havia entrado na casa, passado por todas as portas uma única vez e, em seguida, deixado a casa. Sherlock Gomes avaliou a planta da residência (conforme figura acima) e em poucos minutos declarou solucionado o caso.
Quem poderia ser o suspeito indicado por Sherlock Gomes? Qual o raciocínio utilizado pelo detetive para apontar o suspeito?

Um grafo G é dito ser **hamiltoniano** se existe um ciclo em G que contenha todos os seus vértices, sendo que cada vértice só aparece uma vez no ciclo. Este ciclo é chamado de ciclo hamiltoniano. Sendo assim, um grafo é hamiltoniano se ele contiver um ciclo hamiltoniano.

A título de exemplo, considere os grafos G1 e G2 abaixo. É fácil notar que G1 contém o ciclo (v1, v2, v3, v4, v5, v1) que é hamiltoniano. Logo, G1 é um grafo hamiltoniano. O mesmo não acontece com G2 .



O adjetivo "hamiltoniano" deve-se ao matemático irlandês William Rowan Hamilton (1805-1865).

O problema do ciclo hamiltoniano pode ser resolvido para um dado grafo através de tentativas e erros da seguinte maneira: comece por algum vértice do grafo e tente qualquer caminho escolhendo suas arestas.
Se o caminho resultante tiver um vértice repetido, este caminho deve ser descartado e deve-se testar outro possível caminho.
Este algoritmo envolve algum armazenamento dos caminhos já testados de forma que um mesmo caminho não seja testado mais de uma vez. A abordagem por tentativa e erro, é possível - mas é praticamente inviável em grafos que não forem pequenos, pois existirão muito caminhos a serem testados.

OBS.: Não se conhece na literatura nenhum algoritmo eficiente para se determinar se um ciclo hamiltoniano existe. Alguns autores, dizem que existem algumas evidências que sugerem que este tipo de algoritmo nunca será encontrado.

Embora, não se dispõe de um método conveniente para determinar se um grafo é Hamiltoniano. Há diversos teoremas específicos para determinados tipos de grafos, os quais fornecem condições que são, na maior parte dos casos, suficientes, porém não necessárias.

5.3.2 Problema do Caixeiro Viajante

Suponha que a área de venda de um caixeiro viajante inclua várias cidades, muitas das quais, aos pares, estão conectadas por rodovias. O trabalho do caixeiro requer que ele visite cada cidade pessoalmente. Sob que condições seria possível ele estabelecer uma viagem circular (que o leve ao ponto de partida) de forma a que ele visite cada cidade exatamente uma vez?

Este problema pode ser modelado por um grafo G(V,A), onde:

V = { c | c é uma cidade}

A = { (c1,c2) | há uma estrada que conecta as cidades c1 e c2, sendo que ela não passa por nenhuma outra cidade neste trajeto}

Modelado desta forma, a solução deste problema passa por verificar se o grafo G é hamiltoniano.

Solução: O número de ciclos hamiltonianos pode ser calculado utilizando-se como ferramenta o teorema abaixo.

Teorema: Em um grafo completo, com n vértices, existem (n-1)!/2 ciclos hamiltonianos com arestas disjuntas, se n for ímpar maior que 2, e (n-2)!/2 se n for par.

No entanto, a técnica proposta pelo Teorema acima é muito "cara", pois temos que enumerar todos os ciclos hamiltonianos e escolher aquele de menor custo. Por exemplo, se tivermos 10 vértices em um grafo, teremos 9!/2 = 181440 ciclos a serem testados. Em um grafo com 20 vértices ter-se-á 6E+16 ciclos a serem testados.
Se considerarmos que o computador leve 1E-09 segundos para efetuar cada operação, então seria necessários 6E+16/(10*3*1E+07) = 2 anos para chegar à solução ótima!

E neste caso, não convém o caixeiro esperar uma solução ótima para iniciar sua viagem.

Existem ainda disponíveis alguns métodos com soluções heurísticas que dão um caminho de custo pequeno, mas não garantem que seja o menor, como é o caso do algoritmo a seguir:

5.3.3 Método Algébrico

Este método envolve a geração de todos os caminhos simples por multiplicação sucessiva de matriz. Envolve os seguintes passos:

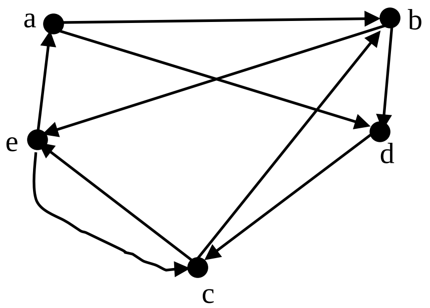
P1. Construa inicialmente a matriz de adjacência A do grafo.
P2. Construa a matriz B(n x n) da seguinte forma:

$$b_{ij} = \begin{cases} v_j, & \text{se existe a aresta } (v_i, v_j) \\ 0, & \text{Caso contrário} \end{cases}$$

P3. Faça $P_1 \leftarrow A$;
P4. Faça $P_{i+1} \leftarrow B * P_i$, i = 1,2...., n-1 onde
$P_{i+1}(k,k) = 0$ para todo k

$$P_{i+1}(s,t) = \boxed{\sum_{k}^{n-1} (b(s,k) * P_i(k,t))}$$

Exemplo:
Determinar os caminhos hamiltonianos do grafo abaixo através do método algébrico:



5.4 Menor Caminho Entre Dois Vértices

Se o problema for determinar o menor caminho entre os vértices vi e vj, sem restrição, então podemos utilizar os algoritmos: de Floyd e Dijkstra.

5.4.1 Algoritmo de Floyd

- É um algoritmo matricial;
- Baseia-se na construção de uma matriz D0 de custos das arestas;
- Para a determinação do caminho, parte-se do final para o início, levando-se em conta os vértices intermediários incluídos durante o processo.

Algoritmo de Floyd

Considere que a matriz de custo foi iniciada de tal modo que $d_{ii} = 0$ para todo i = 1, 2,....n e que $d_{ij} = \propto$, quando não existe a aresta (x_i, x_j). $d_{ij} = C(x_i, x_j)$ se $\exists\ (x_i, x_j)$.

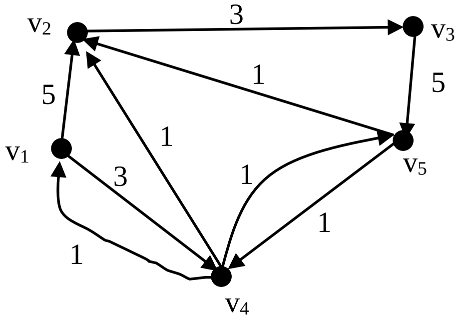
P1. Faça K ← 0;
P2. Faça k ← k+1;
P3. Para todo i ≠ k tal que dik ≠ ∝ e todo j ≠ k tal que dkj ≠ ∝

Faça $$d_{ij}^{k} = \min[d_{ij}^{k-1}, (d_{ik}^{k-1} + d_{kj}^{k-1})]$$

P4. Teste de finalização
a) Se algum d_{ii} >= 0, e k = n, a solução foi achada, e $[d_{ij}]$ fornece os comprimentos de todos os menores caminhos. Pare.
b) Se todo d_{ii} >= 0 mas k < n, então retorne a P2.

Exemplo:

Considere o grafo G(V,A) abaixo, onde os pesos das arestas são considerados aqueles próximos às mesmas. Determinar a menor distância do vértice vi ao vj para todo i, j < n+1, fazendo uso do algoritmo de Floyd.



5.4.2 Algoritmo de Dijkstra (1950)

É outro algoritmo que pode ser utilizado para resolver problemas de caminho mínimo. Seja um grafo G(V,A) e uma função distância L que associe cada aresta (v,w) a um número real não negativo L(v,w) e também um vértice fixo v_0 em V, chamado fonte. O problema consiste em se determinar os caminhos de v_0 para cada vértice v de G, de tal forma que o somatório das distâncias das arestas envolvidas em cada caminho seja mínimo.

Isto é equivalente a determinar um caminho v0, v1,...., vk tal que:

$$\sum_{i=0}^{k-1} L(v_i, v_{i+1}) \quad \text{seja mínimo}$$

Idéia Geral do Algoritmo de Dijkstra

Consideremos o grafo G(V,A), uma fonte v_0 e uma função L que associe cada aresta a um número real não negativo, ou seja:

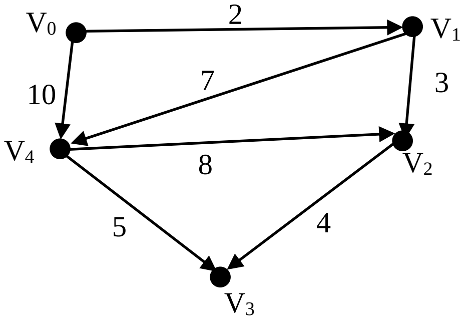
$$L(v_i,v_j)= \begin{cases} \propto, \text{ se } \exists\ (v_i, v_j) \\ 0, \text{ se } v_i = v_j \\ \text{Custo, se } (v_i \neq v_j) \text{ e } \exists\ (v_i, v_j) \end{cases}$$

- Constrói-se um conjunto S, que contém os vértices v_i's cujo comprimento mínimo de v_0 a cada v_i, seja conhecido.
- A cada passo se adiciona ao conjunto S o vértice w pertencente a V - S tal que o comprimento do caminho v_0 a w seja menor do que o correspondente de qualquer outro vértice de V - S.

Algoritmo de Dijkstra

```
Início
        S ← {v0};
        D[v0] ← 0;
        Para cada v ∈ V - {v0} faça D[v] ← L(v0, v);
        Enquanto S ≠ V faça
        Início
                Escolha o vértice w ∈ V - S tal que D[w] seja mínimo;
                Coloque w em S, isto é, faça S ← S U {w};
                Para cada v ∈ V - S faça
                        D[v] ← min [d(v), d(w) + L(w,v)]
        Fim;
Fim.
```

Exemplo:



5.5 Matriz de Roteamento

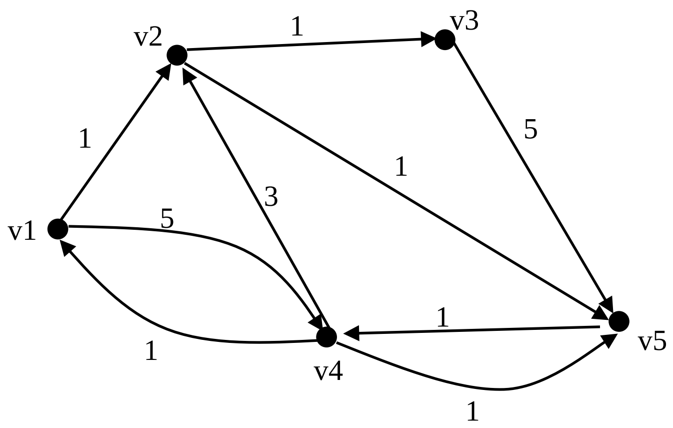
Em diversas situações deseja-se saber qual o menor caminho de um vértice a outro. Uma maneira de se conhecer esse caminho é através da matriz de roteamento.

Baseia-se nos moldes do algoritmo de Floyd.

5.5.1 Algoritmo de Floyd Modificado

```
Entrada: Matriz de Custos D
Saída: A → Matriz com os comprimentos dos menores caminhos
       R → Fornece o vértice k que é o primeiro a ser visitado no menor caminho de vi
           até vj.
Início
       Para i =1 até n Faça
              Para j = 1 até n Faça
                     A[i,j] ← D[i,j];
                     R[i,j] ← 0;
       Para i = 1 até n Faça
              A[i,i] ← 0;
       Para k = 1 até n Faça
              Para i = 1 até n Faça
                     Para j = 1 até n Faça
                            Se A[i,k] + A[k,j] < A[i,j] então
                                   A[i,j] ← A[i,k]+A[k,j];
                                   R[i,j] ← k;
Fim.
```

Exemplo: Considere o grafo abaixo:



A partir da matriz de custo D, aplicando o algoritmo de Floyd Modificado, pode-se obter a matriz A, com os comprimentos dos menores caminhos e a matriz de roteamento R.

$$D = \begin{bmatrix} 0 & 1 & \infty & 5 & \infty \\ \infty & 0 & 1 & \infty & 1 \\ \infty & \infty & 0 & \infty & 5 \\ 1 & 3 & \infty & 0 & 1 \\ \infty & \infty & \infty & 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 2 \\ 3 & 0 & 1 & 2 & 1 \\ 7 & 8 & 0 & 6 & 5 \\ 1 & 2 & 3 & 0 & 1 \\ 2 & 3 & 4 & 1 & 0 \end{bmatrix}$$

$$R = \begin{bmatrix} v1 & v2 & v2 & v5 & v2 \\ v5 & v2 & v3 & v5 & v5 \\ v5 & v5 & v3 & v5 & v5 \\ v1 & v1 & v2 & v4 & v5 \\ v4 & v4 & v4 & v4 & v5 \end{bmatrix}$$

Agora, por exemplo, para determinar a rota de v2 a v1, toma-se R[2,1] = v5, R[5,1]= v4, R[4,1] = v1.

Logo, a rota de v2 a v1 é **v2 - v5 - v4 - v1.** Porém, poderia ter algum vértice intermediário entre os pares de vértices, ou seja, entre “v2 – v5”; “v5 – v4” e “v4 – v1”. Então, deve-se verificar a rota também entre esses pares de vértices. Observa-se na matriz R, neste caso, que não há nenhum vértice intermediário.

Por exemplo, a rota de v2 a v5, toma-se R[2,5] = v5. Logo, v2 está diretamente conectado a v5.

Exercício: Determine a rota de v1 a v4.

REFERÊNCIAS BIBLIOGRÁFICAS

Essas notas de aula foram elaboradas com base nas referências bibliográficas abaixo:

BOAVENTURA NETTO, Paulo Oswaldo. **Teoria e modelos de grafos**. São Paulo: Edgard Blucher, c1979. xi, 249p.

RABUSKE, Marcia Aguiar. **Introdução a teoria dos grafos**. Florianópolis: Ed. da UFSC, 1992.