

Fábio Protti - UFF
Loana T. Nogueira - UFF
Sulamita Klein - UFRJ

Partições em Grafos: Teoria e Aplicações

Motivação

- Suponha que temos um grupo de pessoas (funcionário de uma empresa) que serão submetidos a um treinamento.
 - Queremos identificar os grupos de pessoas que se conhecem (entre si) e os grupos de pessoas que não se conhecem (entre si).

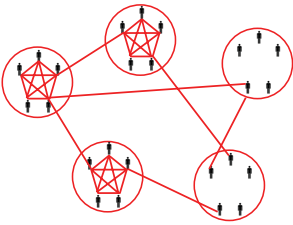


grupo de pessoas que não se conhecem



grupo de pessoas que se conhecem

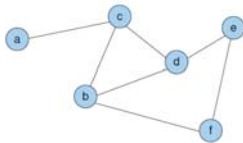
Motivação



qualquer relação entre pessoas de grupos diferentes

Conceitos Básicos

- O que é um grafo?
 - É um conjunto de pontos, chamados vértices, conectados por linhas, chamadas de arestas.



É um grafo?

Conceitos Básicos

- Abstração que permite codificar relacionamentos entre pares de objetos

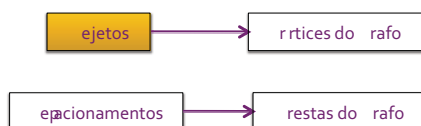
ua is oejetosç

- Ex.: Pessoas, cidades, empresas, países, páginas web, filmes, etc.

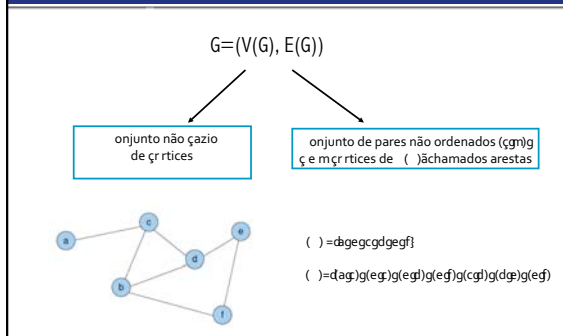
ua is reaçionamentosç

- Ex.: Amizade, conectividade, produção, língua falada, etc.

Grafos – Definição Informal



Definição



Definições e Conceitos

- Os extremos de uma aresta são ditos **incidentes** com a aresta, e vice-versa.

Ex.: u e v são incidentes a e

- Dois vértices que são incidentes a uma mesma aresta são ditos **adjacentes**.

Ex.: u e v são adjacentes

Definições e Conceitos

- Conjunto vizinhança

conjunto vizinhança de um vértice u é

$N(u) = \{v \mid uv \in E\}$ dos vértices adjacentes a u



Classes especiais de grafos

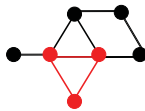
- **Grafo Completo:** grafo em que cada par de vértices distintos possui uma aresta.



gráfico completo com 5 vértices

Clique

um subconjunto de vértices
que induz um subgrafo completo



Classes especiais de grafos

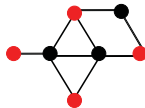
- **Grafo Vazio:** é um grafo sem arestas.



gráfico vazio com 5 vértices

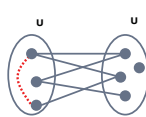
Conjunto Independente

um subconjunto de vértices
que induz um subgrafo vazio



Classes especiais de grafos

- **Grafo Bipartido:** é aquele em que o conjunto de vértices pode ser particionado em dois subconjuntos X e Y , tal que cada aresta tem um extremo em X e um em Y .

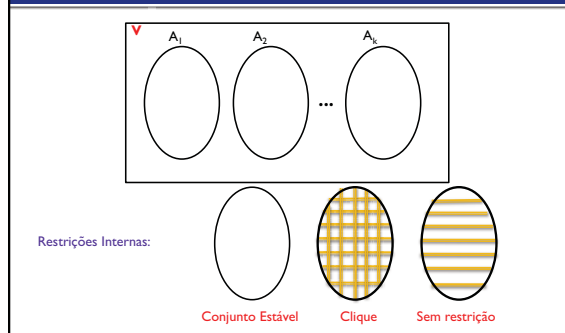


(X, Y) é uma bipartição
do grafo

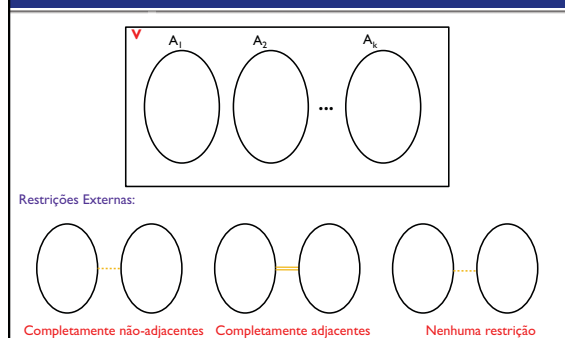
Classes especiais de grafos

- **Grafo Bipartido Completo:** é um grafo bipartido com bipartição (X, Y) em que cada vértice de X é adjacente a cada vértice de Y .
- Se $|X|=m$ e $|Y|=n$, então denotamos tal grafo por $K_{m,n}$

Problema de Partição de Grafos



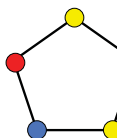
Problema de Partição de Grafos



Exemplo: o problema da 3-coloração

Entrada: Grafo G

Objetivo: Determinar o menor número de cores necessárias para colorir os vértices de G de forma que vértices adjacentes recebam cores diferentes



que podemos afirmar sobre os vértices com uma mesma cor?

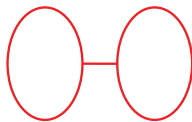
Induzem um subgrafo próprio (conj. estável)

Problema da bipartição: determinar o menor t tal que G pode ser particionado em t conjuntos independentes.

Exemplo: Grafo Bipartido

Entrada: Grafo G

Objetivo: Determinar se G pode ser particionado em dois conjuntos independentes



Se G é bipartido



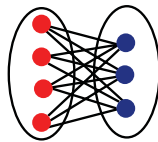
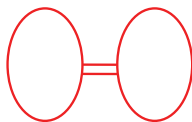
G pode ser colorido com 2 cores

2-coloração

Exemplo: Grafo Bipartido Completo

Entrada: Grafo G

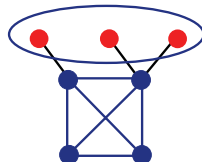
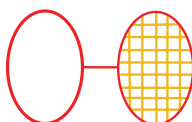
Objetivo: Determinar se G pode ser particionado em dois conjuntos independentes completamente adjacentes



Exemplo: Grafo Split

Entrada: Grafo G

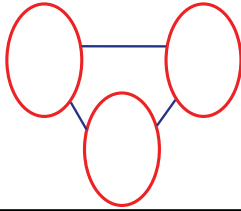
Objetivo: Determinar se G pode ser particionado em um conjunto independente e em uma clique



Exemplo: 3-coloração

Entrada: Grafo G

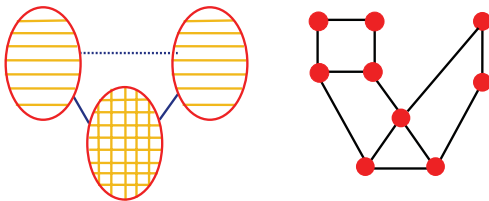
Objetivo: Determinar se G pode ser particionado em três conjuntos independentes



Exemplo: Corte-clique

Entrada: Grafo conexo G

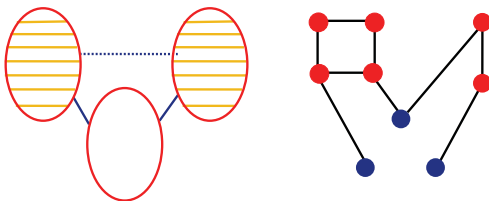
Objetivo: Determinar se G possui uma clique C tal que $G-C$ seja desconexo



Exemplo: Corte-estável

Entrada: Grafo conexo G

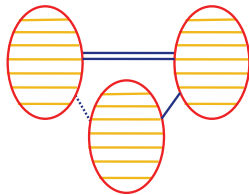
Objetivo: Determinar se G possui um conjunto estável S tal que $G-S$ é desconexo



Exemplo: Partição Módulo

Entrada: Grafo conexo G

Objetivo: Particionar G em 3 conjuntos: A , N , M quaisquer tais que:



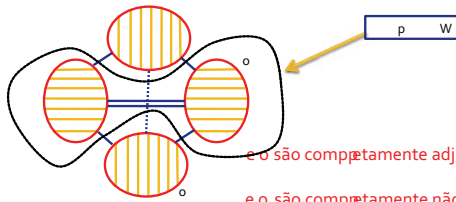
A e M são completamente adjacentes

N e M são completamente não adjacentes

Exemplo: Partição Skew

Entrada: Grafo conexo G

Objetivo: Particionar G em 4 conjuntos: A , B , C , D tais que:



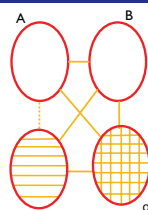
A e B são completamente adjacentes

C e D são completamente não adjacentes

Exemplo: Partição Teimosa (Stubborn)

Entrada: Grafo conexo G

Objetivo: Particionar G em 4 conjuntos: A , B , C , D tais que:



A e B são conjuntos independentes

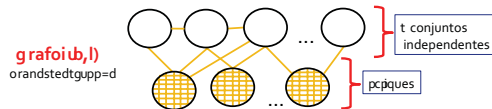
C e D são conjuntos independentes

A e C são completamente não adjacentes

Partição-(k,l)

Entrada: Grafo G
Objetivo: Particionar G em k conjuntos independentes e l cliques

- k conjuntos independentes: $V_1, V_2, \dots, V_k$
- l cliques: $V_{k+1}, V_{k+2}, \dots, V_{k+l}$
- Não há restrição entre cada um dos conjuntos



Grafos-(k,l)

ulagguuu g rafog	b	l
rafo azio	u	rafo-(ug)
rafo ompæto	u	rafo-(g)
rafo ppit	u u	rafo-(ug)
rafo eipartido		rafo-(g)
rafo co=eipartido		rafo-(g)
orandstedtupp=d	u	rafo-(ug)
orandstedtupp=d	u	rafo-(g)
orandstedtupp=d		rafo-(g)
co=opração		rafo-(g)
co=opração		rafo-(g)
rafo-(t g)		r=compæto

Problema Geral da Partição

Entrada: Grafo G
Objetivo: É possível particionar $V(G)$ em p subconjuntos (partes): $V_1, V_2, \dots, V_p$ tais que:

- V_i { Clique
Conjunto Independente
Não possui restrição
- V_i, V_j { Completamente adjacentes
Completamente não adjacentes
Não possui restrição
- estrições internas*
- estrições externas*

Problema da M-partição

Seja M uma matriz simétrica definida sobre $\{0, 1, *\}$

Uma M -partição de G é uma partição $V = V_1, V_2, \dots, V_p$,
 $V_i \cap V_j = \emptyset, i \neq j, i, j \in \{1, 2, \dots, p\}$, tal que

- M_{ii} $\left\{ \begin{array}{l} 1 \text{ se } V_i \text{ é uma clique} \\ 0 \text{ se } V_i \text{ é um conjunto independente} \\ * \text{ se } V_i \text{ não possui restrição} \end{array} \right.$ diagonal principal
↓
restrições internas
- M_{ij} $\left\{ \begin{array}{l} 1 \text{ se } V_i, V_j \text{ são completamente adjacentes} \\ 0 \text{ se } V_i, V_j \text{ são completamente não adjacentes} \\ * \text{ se } V_i, V_j \text{ não possui restrição} \end{array} \right.$ sem. fora diagonal principal
↓
restrições externas

Problema da M-partição

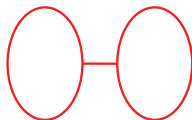
Entrada: Grafo G

Objetivo: G possui uma M -partição com respeito à matriz M ?

Exemplo: Grafo Bipartido

Entrada: Grafo G

Objetivo: Determinar se G pode ser particionado em dois conjuntos independentes

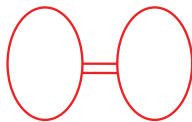


$$= \begin{bmatrix} & a \\ a & \end{bmatrix}$$

Exemplo: Grafo Bipartido Completo

Entrada: Grafo G

Objetivo: Determinar se G pode ser particionado em dois conjuntos independentes completamente adjacentes

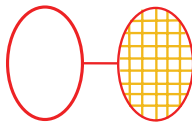


$$= \begin{pmatrix} u & u \\ u & u \end{pmatrix}$$

Exemplo: Grafo Split

Entrada: Grafo G

Objetivo: Determinar se G pode ser particionado em um conjunto independente e em uma clique

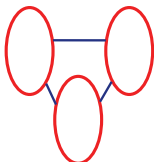


$$= \begin{pmatrix} u & a \\ a & a \end{pmatrix}$$

Exemplo: 3-coloração

Entrada: Grafo G

Objetivo: Determinar se G pode ser particionado em três conjuntos independentes

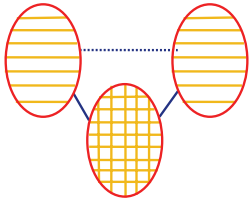


$$= \begin{pmatrix} & a & a \\ a & & a \\ a & a & \end{pmatrix}$$

Exemplo: Corte-clique

Entrada: Grafo conexo G

Objetivo: Determinar se G possui uma clique C tal que G-C seja desconexo

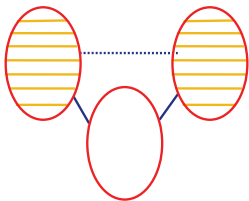


$$= \begin{pmatrix} a & u & a \\ a & u & a \\ a & a & a \end{pmatrix}$$

Exemplo: Corte-estável

Entrada: Grafo conexo G

Objetivo: Determinar se G possui um conjunto estável S tal que G-S é desconexo

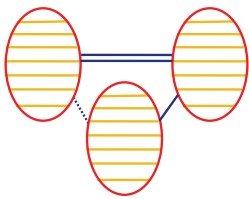


$$= \begin{pmatrix} a & a \\ a & a \\ a & a \end{pmatrix}$$

Exemplo: Partição Módulo

Entrada: Grafo conexo G

Objetivo: Particionar G em 3 conjuntos: A, N, M quaisquer tais que:

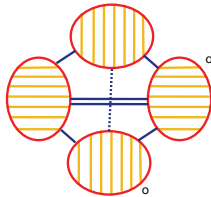


$$= \begin{pmatrix} a & u & a \\ a & u & a \\ a & a & a \end{pmatrix}$$

Exemplo: Partição Skew

Entrada: Grafo conexo G

Objetivo: Particionar G em 4 conjuntos: A, B, C, D tais que:

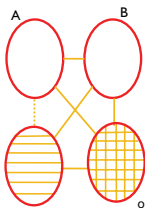


$$= \begin{pmatrix} a & u & a & a \\ u & a & a & a \\ a & a & a & a \\ a & a & a & a \end{pmatrix}$$

Exemplo: Partição Teimosa (Stubborn)

Entrada: Grafo conexo G

Objetivo: Particionar G em 4 conjuntos: A, B, C, D tais que:

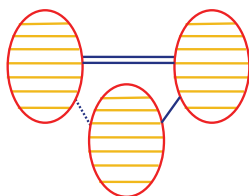


$$M = \begin{pmatrix} A & B & C & D \\ 0 & * & 0 & * \\ * & 0 & * & * \\ 0 & * & * & * \\ * & * & * & I \end{pmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Retornando ao problema Partição Módulo

Entrada: Grafo conexo G

Objetivo: Particionar G em 3 conjuntos: A, N, M quaisquer tais que:



m partição r dito um conjunto homoganeo se

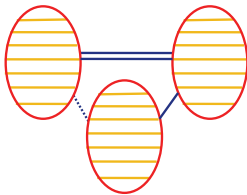
$$WW \text{ e } WU \text{ Wu}$$

ugtrição uu paruinaiuuu
nogponéntog

Ex.: Conjunto Homogêneo

Entrada: Grafo conexo G , matriz M e $L(v)$: lista de partes a qual v pode pertencer

Objetivo: G admite partição com respeito a M e a $L(v)$?



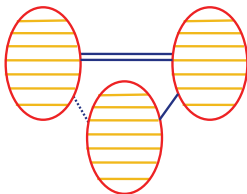
$WW \in WU \setminus Wv$
 $sco\phi a \quad \phi r tices\tilde{a}, g\tilde{i} \text{ e } z$
 $(.) = d \}$
 $\{\} = d \}$
 $(z) = d \setminus g \}$

n^3 problemas de partição

Ex.: Conjunto Homogêneo

Entrada: Grafo conexo G , matriz M e $L(v)$: lista de partes a qual v pode pertencer

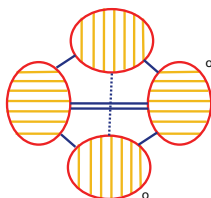
Objetivo: G admite partição com respeito a M e a $L(v)$?



possui um conjunto homoganeo
 $\Updownarrow$
 uma das n poss-zeis
 escolhas de $. g\tilde{i} \text{ e } z$ nos dq
 um conjunto homoganeo

Ex.: Partição Skew

Como resolver o problema para partição skew?



$gog \setminus go \setminus s \setminus \emptyset$



Fábio Protti - UFF
Loana T. Nogueira - UFF
Sulamita Klein - UFRJ

Parte 2: Algoritmos para problemas de partição em grafos

Algoritmo Conjunto Independente

Entrada: Grafo G
Objetivo: Testar se G é $(1,0)$

- Descrição:
 - O algoritmo é trivial.
 - Basta verificar se o número de arestas é zero

Complexidade: $O(n+m)$ - Linear

Algoritmo Completo

Entrada: Grafo G
Objetivo: Testar se G é $(0, 1)$

- Descrição:
 - O algoritmo é trivial.
 - Basta verificar se o número de arestas é igual a $n(n-1)/2$ (i.e., se $V(G)$ é uma clique)

Complexidade: $O(n+m)$ - Linear

Algoritmo Bipartido

Entrada: Grafo G

Objetivo: Testar se G é (2, 0)

■ Descrição:

- O algoritmo testa se G é 2-colorível (bipartido) através de uma busca em profundidade.
- Em caso afirmativo, devolve uma 2-coloração
- Em caso negativo, devolve um ciclo ímpar

Algoritmo Bipartido

Entrada: Grafo G

Objetivo: Testar se G é (2, 0)

■ Algoritmo_BIPARTIDO

Cor := 0;

Desmarcar todos os vértices;

Enquanto existe vértice v desmarcado faça

BICOLORAÇÃO(v)

Algoritmo Bipartido

■ Algoritmo_BICOLORAÇÃO

Marcar v;

Cor(v) := COR;

COR := 1-COR;

Para todo vizinho w de v faça

Se w está desmarcado então

BICOLORAÇÃO(w)

Senão Se Cor(v) = Cor(w) então

Devolver ciclo ímpar contendo v, w

Algoritmo Bipartido

Entrada: Grafo G

Objetivo: Testar se G é $(0, 1)$

■ **Descrição:**

- O algoritmo testa se G é 2-colorível (bipartido) através de uma busca em profundidade.
- Em caso afirmativo, devolve uma 2-coloração
- Em caso negativo, devolve um ciclo ímpar

Complexidade: $O(n+m)$ - Linear

Consequência do Algoritmo Bipartido

G é bipartido



G não contém ciclo ímpar

Algoritmo Co-Bipartido

Entrada: Grafo G

Objetivo: Testar se G é $(0, 2)$

■ **Descrição:**

- Como G é $(0,2)$ se e somente se $\overline{G}$ é $(2,0)$, basta executar o algoritmo Bipartido no grafo $\overline{G}$.

Complexidade: $O(n+\overline{m})$, onde $\overline{m} = n(n-1)/2 - m$

Algoritmo Bipartido Completo

Entrada: Grafo G

Objetivo: Testar se G é bipartido completo

Algoritmo BIPARTIDO_COMPLETO

Executar o algoritmo Bipartido.

p := número de vértices com cor 0

q := número de vértices com cor 1

Se $m = pq$ então G é bipartido completo

Complexidade: $O(n+m)$ - Linear

Alguns Fatos sobre Grafos Split

Todo grafo split é cordal

Um grafo cordal é aquele que não possui ciclo induzido de tamanho maior ou igual 4.



G é cordal se e somente se admite um esquema de eliminação perfeita (EEP)

Alguns Fatos sobre Grafos Split

Um EEP é uma ordenação $v_1, v_2, \dots, v_n$ de $V(G)$ tal que:

Para todo i , v_i e seus vizinhos $v_j, j > i$ formam uma clique



formam uma clique

Um EEP pode ser encontrado em tempo linear

Algoritmo Split

Entrada: Grafo G

Objetivo: Testar se G é (1,1)

Descrição:

- Algoritmo baseia-se em percorrer um EEP atribuindo cores 0 ou 1 aos vértices
- Se G for split, o algoritmo exibe uma coloração onde:
 - vértices com cor 0 $\rightarrow$ conjunto independente
 - Vértices com cor 1 $\rightarrow$ clique

Algoritmo Split

Entrada: Grafo G

Objetivo: Testar se G é (1,1)

Descrição:

- Se G não for split, o algoritmo exibe um certificado de que o grafo não é split (obstrução")



Algoritmo Split

O Algoritmo consiste de 3 etapas:

- Encontre um EEP de G
 - Se não existir então PARE: G não é Split. Exiba "obstrução.
 - Senão $cor(v_1) := 0$, e continue usando a cor 0 para $v_2, v_3, \dots, v_j$, para o maior j possível
 - Se $j=n$ então PARE (G é split);
 - Senão
 - $cor(v_{j+1}) := 1$.

Algoritmo Split

O Algoritmo consiste de 3 etapas:

- Se $j+1=n$ então PARE (G é Split)
- Senão Para $i=j+2, j+3, \dots, n$
 - Atribua cor a v_i da seguinte forma:
 - (i) Se v_i é adjacente a v_{j+1} então $cor(v_i) := 1$
 - (ii) Se v_i não é adjacente a v_{j+1} então verifique se v_i é não adjacente a todos os vértices já coloridos com a cor 0; nesse caso, faça $cor(v_i) = 0$.
 - (iii) Se (i) e (ii) não se aplicam a v_i , então PARE: G não é split. Exiba "obstrução"

Complexidade: $O(n+m)$ - Linear

Algoritmo Split

Teorema: G é split se e somente se G não contém C_4 , C_5 e $2K_2$ como subgrafo induzido

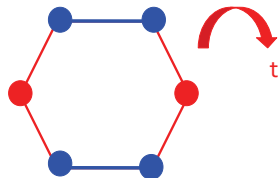
- Se o algoritmo split para na etapa 1:

encontre um r de
e não e. istirentão r $\bar{a}$ não r pít

- Encontramos um C_4 , C_5 ou C_k ($k \geq 6$)

C_k ($k \geq 6$): contém um $2K_2$

..ã



Algoritmo Split

Teorema: G é split se e somente se G não contém C_4 , C_5 e $2K_2$ como subgrafo induzido

- Se o algoritmo split para na condição 3.(iii), existe $2K_2$:

enão r ara i= jr gjr g. gñ

trieva cor a ç_i da seguinte formaã

(i) e ç_i r adjacente a ç_j r, então cor(ç_i)=a u

(ii) e ç_i não r adjacente a ç_j r, então ç_i geriz que se ç_i r não adjacente a todos os ç_r r tices jq copridos com a cor ; nesse casogfaça cor(ç_i)= .

(iii) e (i) e (ii) não se apcam a ç_i g então r ã não r spñt. .

. iea doestrução d

[illegible]

Algoritmo (2,1)

Entrada: Grafo G

Objetivo: Testar se G é $(2,1)$

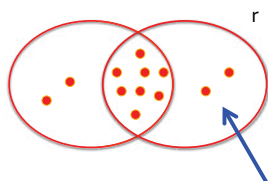
Descrição:

- O algoritmo baseia-se no seguinte fato:
 - Se $B \subseteq V(G)$ induz um subgrafo bipartido e $C \subseteq V(G)$ é uma clique então $|B \cap C| \leq 2$
 - Logo, se existir uma $(2,1)$ -partição de G utilizando uma clique C (que supomos ser a maior possível nessas condições), qualquer clique C'' de G satisfaz

$$|C \cap C'| \geq |C'| - 2$$

[illegible]

Algoritmo (2,1)



ão pode fazer cr rtices em m
senão e. istiria trit nqup



Algoritmo (2,1)

Descrição:

- Fase 0: Encontrar clique inicial C'
- Fase 1: aumentar C' até o tamanho máximo possível
 - **Repita**
 - Fazer trocas (busca local aumentante) de 1 vértice de C' por 2 vértices fora de C' , ou de 2 vértices de C' por 3 vértices fora de C' , até que não seja mais possível
 - **Até que** não seja mais possível aumentar C'

Algoritmo (2,1)

Descrição:

- Fase 2: testar (2,1)-partições
 - Fazer trocas de 1 vértice de C' por 1 vértice fora de C' , ou de 2 vértices de C' por 2 vértices fora de C' , testando sempre se $(V(G) \setminus C', C')$ é uma (2,1)-partição

Algoritmo (2,1)

Complexidade: $O(n^6m)$ para Fase 1
 $O(n^4m)$ para Fase 2
Total: $O(n^6m)$

Complexidade: [Brandstädt, 96,98] — $O(n^3)$

Algoritmo Partições Esparso-denso [Feder,Hell, K, Motwani, 1999]

Entrada: Grafo G
Objetivo: Testar se $V(G)$ pode ser particionado em um subconjunto "esparso" S e um subconjunto "denso" D

Dizemos que duas classes S e D formam um par esparso-denso se existe uma constante c tal que $|S \cap D| \leq c, \forall S \in \mathcal{S}, \forall D \in \mathcal{D}$.

Exemplo de Pares Esparso-denso

induz...	o induz...	$W \cap W^c$
grafo sem arestas	grafo completo	$c=0$
grafo bipartido	grafo completo	$c=1$
grafo estrela	grafo completo	$c=1$
grafo ciclo de comprimento ímpar	grafo completo	$c=1$
grafo bipartido	grafo co-bipartido	$c=1$
grafo planar	grafo completo	$c=1$
grafo acíclico	completo de n vértices	$c=n-1$

Algoritmo Partições Esparso-denso [Feder,Hell, K, Motwani, 1999]

- Descrição:**
- Fase 0: encontrar esparso inicial S'
 - Fase 1: busca local aumentante:
 - para fazer trocas de c' vértices de S por $c'+1$ vértices fora de S , com $c' \leq c$
 - Fase 2: testar partições esparso-denso:
 - Fazer trocas de c' vértices de S por c' vértices fora de S , com $c' \leq c$, testando se $(S', D' = V(G) \setminus S')$ é uma partição esparso-denso

$(S, D = V(G) \setminus S)$



Algoritmo (2,2)

Entrada: Grafo G

Objetivo: Testar se G é $(2, 2)$

■ **Descrição:**

■ Use o algoritmo Esparso-Denso, onde:

- $\mathcal{S}$ é a classe dos grafos bipartidos;
- $\mathcal{D}$ a classe dos co-bipartidos.

Complexidade: $O(n^{10}m)$
 $O((n+m)^2)$: [Brandstädt 96,98]

Algoritmos para problemas da Partição- (k,l)

b	l	u	o	m	p	l	u	u	u
u	r	o	j	n	o	m	i	a	p
u	r	o	j	n	o	m	i	a	p
u	u	r	o	j	n	o	m	i	a
u	r	o	j	n	o	m	i	a	p
u	r	o	j	n	o	m	i	a	p
u	r	o	j	n	o	m	i	a	p

E para k ou $l \geq 3$?

Teorema: [Brandstädt, 96/98]
Se $k \geq 3$ ou $l \geq 3$, o problema de testar se G possui uma (k,l) -partição é NP-completo

Algoritmo Corte Clique

Entrada: Grafo G

Objetivo: Testar se G possui um corte clique

■ **Descrição:** [Whitesides, 81]

■ $S :=$ ciclo sem cordas de comprimento ≥ 4

■ **Repita**

- Ache uma componente conexa C de $G-S$
- $R :=$ vizinhança de C em S
- Se R é uma clique então PARE
 - **Senão**
 - $x, y \leftarrow$ vértices não adjacentes em R
 - $p \leftarrow$ caminho sem cordas de x a y com vértices internos de C
 - $S \leftarrow S \cup V(p)$

■ Até que $S=V(G)$

Algoritmo Corte Clique

Fato: k é corte clique $\Leftrightarrow S$ está contido em uma componente conexa de $G-k$.

Complexidade: $O(n.m)$

Corte Clique

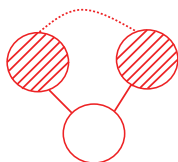
Observação: testar existência de corte clique é um problema de M-partição com partes não vazias.

$$= \begin{pmatrix} 1 & \text{XXXXXX} \\ \text{XXXXXX} & \text{XXXXXX} \\ * & \text{XXXXXX} \end{pmatrix}$$

= m partes não vazias

Outros exemplos: Corte estável

NP-completo: [Brandstädt, Dragan, Le, Szymczak, 2000]

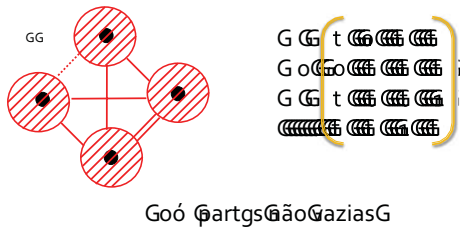


$$= \begin{pmatrix} o & \text{t} & \text{t} \\ \text{t} & \text{t} & \text{t} \\ \text{t} & \text{t} & \text{t} \end{pmatrix}$$

para apagar = $c = r^e$, as partes deverão ser não vazias

Outros exemplos: Partição Skew

Complexidade: $O(n^4m)$ [Kennedy & Reed, 2008]



Problemas de M-partição com listas

Restrições especiais nas partes podem ser modeladas com Problemas de M-Partição com listas

Recordando: Fixada uma matriz M e dado um grafo G com listas, onde para cada vértice $v \in V(G)$ existe uma lista $L(v)$ de partes permitidas para v , é possível encontrar uma M -partição de G que satisfaça as restrições dadas pelas listas?

Algoritmo: Modelando um problema de M-partição em partes não vazias como um problema de M-partição com listas

Fato: O problema de partição com partes não vazias tem solução se e somente se algum dos problemas de partição com listas tem solução

- Complexidade: $O(n^k) \cdot T(n)$, onde
 - $T(n)$ é a complexidade do problema de partição com listas correspondente
- Obs.: Em geral, esta técnica não gera bons algoritmos (ex. Conjunto homogêneo).

Mas ...

Algoritmo: Modelando um problema de M-partição em partes não vazias como um problema de M-partição com listas

Se o problema de M-partição com listas pode ser resolvido em tempo polinomial



O Problema de M-partição em partes não vazias pode ser resolvido em tempo polinomial

Teorema: [Feder, Hell, K., Motwani, 1999]

Se M é uma matriz 2×2 , então o problema da M -partição com listas pode ser resolvido em tempo polinomial

- Dem.: Solução via 2-sat ($v_A = 1$ sss $v \in A$)

Ex.: $\begin{matrix} & \text{GG} & \text{GG} \\ \text{GG} & 0 & 1 \\ \text{GG} & 1 & 1 \end{matrix}$

Teorema: [Feder, Hell, K., Motwani, 1999]

- Dem.: Solução via 2-sat ($v_A = 1$ sss $v \in A$)

Ex.: $\begin{matrix} & GG & GG \\ GG & 0 & 1 \\ GG & 1 & 1 \end{matrix}$

$$\begin{aligned} G(L) &= \{L\}L \\ L(L) &= \{L, LL\}L \\ L(v) &= \{LL\}L \\ L(L) &= \{LL, LLL\}L \end{aligned}$$
[illegible]

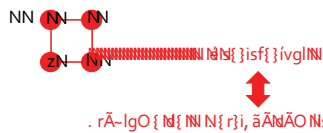
Teorema: [Feder, Hell, K., Motwani, 1999]

- Dem.: Solução via 2-sat ($v_A = 1$ sss $v \in A$)

Ex.:

$$\begin{matrix} & NN & NN \\ NN & \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} N\bar{A}\bar{A}N\bar{N} \\ N\bar{A}\bar{A}N\bar{N} \\ N\bar{A}\bar{A}N\bar{N} \\ N\bar{A}\bar{A}N\bar{N} \end{matrix}$$



Teorema: [Feder, Hell, K., Motwani, 1999]

- Se M é uma matriz 3×3 , então o problema da M -partição com listas é:

- NP-completo nos seguintes casos:

$$M = \begin{pmatrix} 0 & * & * \\ * & 0 & * \\ * & * & 0 \end{pmatrix}, \quad N\bar{A}\bar{A}N\bar{N}, \bar{A}\bar{A}N$$

$$M = \begin{pmatrix} 1 & * & * \\ * & 1 & * \\ * & * & 1 \end{pmatrix}, \quad O\bar{N}g\bar{u}N\bar{A}O, \bar{I}gO\bar{g}N\bar{A}N$$

Teorema: [Feder, Hell, K., Motwani, 1999]

- Se M é uma matriz 3×3 , então o problema da M -partição com listas é:

- NP-completo nos seguintes casos:

$$M = \begin{pmatrix} 0 & * & * \\ * & * & 0 \\ * & 0 & * \end{pmatrix}, \quad N\bar{A}\bar{A}N\bar{N}, \bar{A}\bar{A}N$$

$$M = \begin{pmatrix} 1 & * & * \\ * & * & 1 \\ * & 1 & * \end{pmatrix}, \quad O\bar{N}g\bar{u}N\bar{A}O, \bar{I}gO\bar{g}N\bar{A}N$$

Teorema: [Feder, Hell, K., Motwani, 1999]

- Se M é uma matriz 4×4 , sem $*$'s na diagonal principal, então o problema M -partição com listas é:
 - NP-completo se M contém a matriz correspondente à 3-coloração ou seu complemento
 - Solucionável em tempo polinomial nos demais casos

Corolário: [Feder, Hell, K., Motwani, 1999]

- Se M é uma matriz 4×4 então o problema da M -partição SEM listas é
 - NP-completo se M contém a matriz correspondente à 3-coloração ou seu complemento, e nenhuma entrada diagonal principal é $*$
 - Solucionável em tempo polinomial nos demais casos

Corolário: [Feder, Hell, K., Motwani, 1999]

- Dem.:
 - Observe que n problema sem listas, se houver algum $*$ na diagonal principal então o problema é trivial
 - Basta colocar todos os vértices do grafo em uma parte marcada com $*$ em M .



Corolário: [Feder, Hell, K., Motwani, 1999]

- Se M (de tamanho arbitrário) não contém nenhuma entrada “*”, então o problema da M -partição com listas é solucionável em tempo polinomial.
- Dem.:
 - Suponha que a diagonal de M tenha k “0’s e l “1’s.
 - Um subgrafo formado por k conjuntos independentes não podem conter uma clique com $l+1$ vértices,
 - Um subgrafo formado por l cliques não pode conter um conjunto estável contendo $k+1$ vértices.
 - Use o argumento Esparso-denso para cada uma delas e chegamos à existência da M -partição
