Aula 15 - Emparelhamentos

Luís Felipe

UFF

26 de Maio de 2026

Luís Felipe

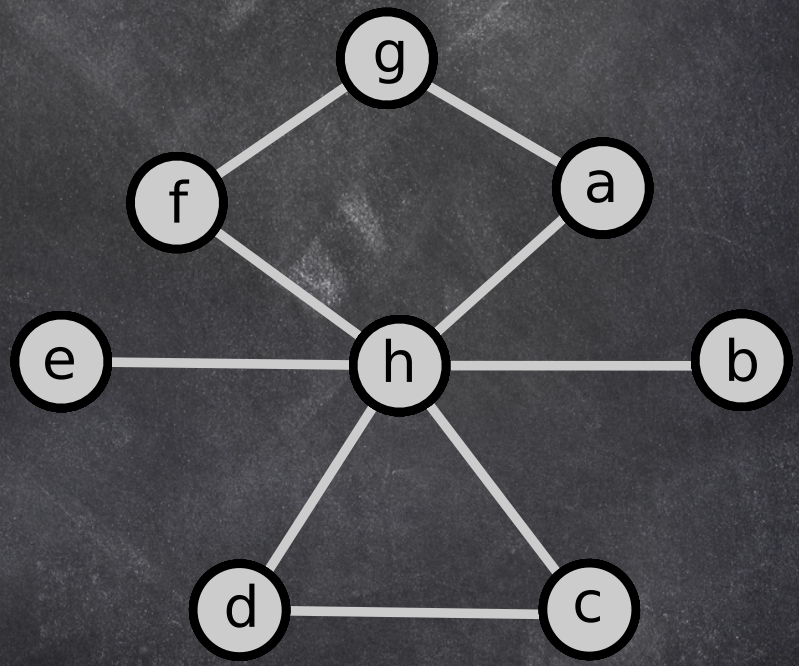
26/05/26

Emparelhamentos

Um emparelhamento $M \subseteq E$ é um subconjunto das arestas tal que qualquer par de arestas em M são não adjacentes (não tem extremo comum).

Exemplo:

- $M_1 = \{fg\}$
- $M_2 = \{fg, ah, cd\}$
- $M_3 = \{eh, cd, ag\}$
- $M_4 = \{ch, fg\}$



Luís Felipe

26/05/26

Emparelhamentos

- Um emparelhamento é maximal se não está contido propriamente em nenhum outro.
- Um emparelhamento é máximo se tem cardinalidade máxima.
- Um emparelhamento é perfeito se tem cardinalidade $\frac{n}{2}$.

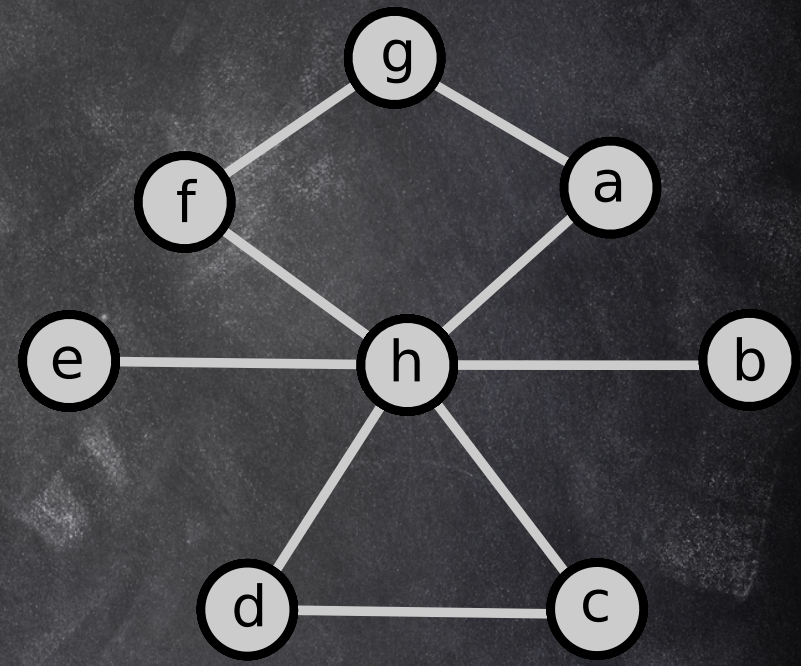
Luís Felipe

26/05/26

Voltando ao exemplo

Exemplo:

- $M_1 = \{fg\}$
- $M_2 = \{fg, ah, cd\}$, máximo
- $M_3 = \{eh, cd, ag\}$, máximo
- $M_4 = \{ch, fg\}$, maximal



Luís Felipe

26/05/26

Emparelhamentos

- Dada uma aresta $xy \in M$, dizemos que os extremos x e y são M-emparelhados.
- Se $\exists e \in M$ incidente a um vértice x, dizemos que x está M-saturado. Caso contrário, dizemos que x é M-insaturado.

Problema alvo: Encontrar um emparelhamento máximo (Problema de otimização)

Ferramentas: caminho alternante
caminho aumentante

Luís Felipe

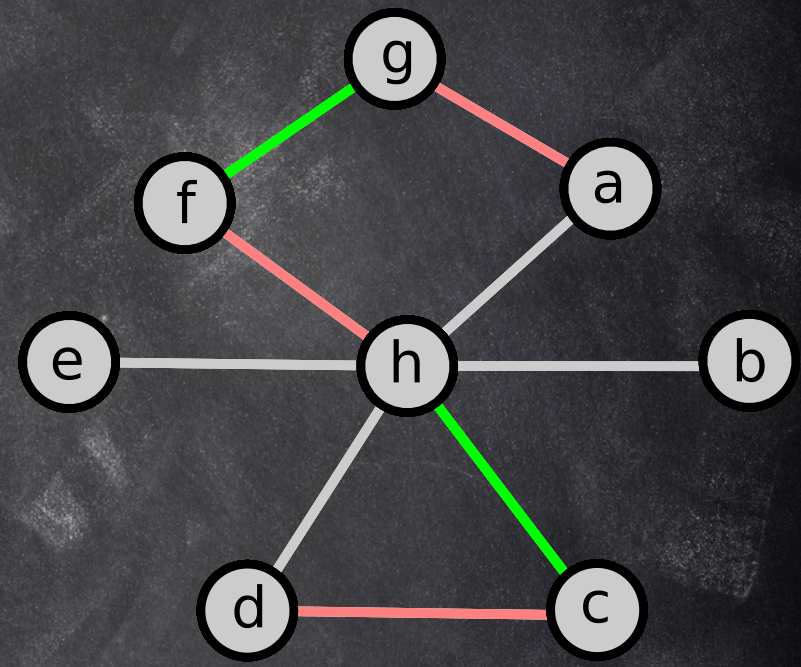
26/05/26

Voltando ao exemplo

Um caminho M-alternante em G é um caminho cujas arestas estão, alternadamente, em M e em $E \setminus M$.

Se, além disso, a origem e o término do caminho são M-insaturados, então o caminho é M-aumentante.
$M_4 = \{ch, fg\}$ é maximal
Observe que $G \setminus \{c, h, f, g\}$ é conjunto independente

O caminho $dchfga$ é M_4-alternante.
Os vértices d, a são M_4-insaturados
Logo, o caminho é M_4-aumentante.



Luís Felipe

26/05/26

Teorema de Berge

Teorema: Seja G um grafo. M é um emparelhamento máximo de G se, e somente se, G não contém caminho M-aumentante.

Prova: ($\rightarrow$) (Prova pela contrapositiva). Suponha que G possui um caminho M-aumentante P. Obtemos o emparelhamento M' tal que $|M'| = |M| + 1$:

- Mantenha as arestas de M que não pertencem a P, seja x esse número de arestas;
- Remova $M \cap E(P)$, ou seja, remova de M as arestas de P;
- Adicione $E(P) \setminus M$, ou seja, adicione as arestas de P que não pertenciam a M.

Todo caminho M-aumentante tem, necessariamente, quantidade par de vértices, pois as arestas alternam entre não pertence e pertence a M, iniciando e finalizando com arestas que não pertencem a M.
Seja $P = v_1, v_2, \ldots, v_{2k-1}, v_{2k}$. Assim, $|M| = x + k - 1$ e $|M'| = x + k$.
Assim, M não é máximo, pois M' tem cardinalidade maior.

Luís Felipe

26/05/26

Para a volta . . .

Para mostrar a volta, vamos dar uma pausa no Teorema de Berge e falar sobre diferença simétrica.

Luís Felipe

26/05/26

Diferença simétrica

Sejam G e H grafos com mesmo conjunto de vértices V. A diferença simétrica $G \triangle H$ é o grafo com conjunto de vértices V cujas arestas são aquelas que aparecem em exatamente um dos grafos.

Considerando conjunto de arestas, em particular, emparelhamentos M e M', $M \triangle M' = (M - M') \cup (M' - M)$.

Exemplo:

Luís Felipe

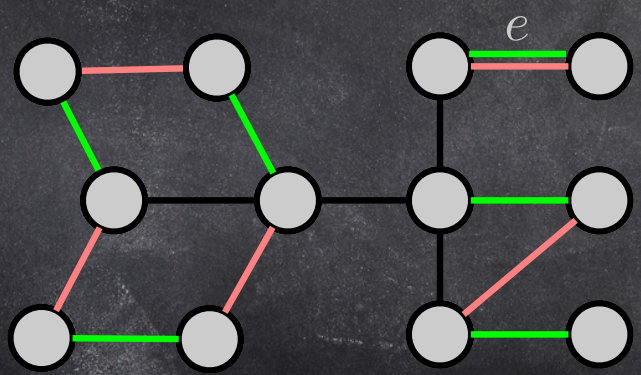
26/05/26

Diferença simétrica

Sejam G e H grafos com mesmo conjunto de vértices V. A diferença simétrica $G \triangle H$ é o grafo com conjunto de vértices V cujas arestas são aquelas que aparecem em exatamente um dos grafos.

Considerando conjunto de arestas, em particular, emparelhamentos M e M', $M \triangle M' = (M - M') \cup (M' - M)$.

Exemplo:



Luís Felipe

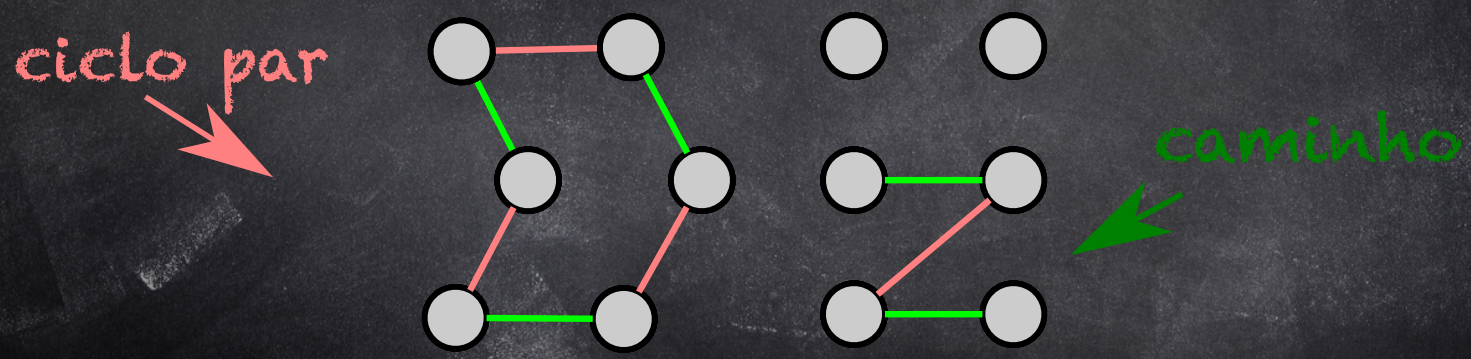
26/05/26

Diferença simétrica

Sejam G e H grafos com mesmo conjunto de vértices V. A diferença simétrica $G \triangle H$ é o grafo com conjunto de vértices V cujas arestas são aquelas que aparecem em exatamente um dos grafos.

Considerando conjunto de arestas, em particular, emparelhamentos M e M', $M \triangle M' = (M - M') \cup (M' - M)$.

Exemplo:



Luís Felipe

26/05/26

Lema (intermediário ao Teo de Berge)

Lema: Cada componente conexa da diferença simétrica de dois emparelhamentos é ou um ciclo par ou um caminho.

Prova: Cada vértice de M ou M' possui grau no máximo 2 em $M \Delta M'$, pois cada vértice é extremo de no máximo uma aresta de cada emparelhamento. Com isso, cada componente é um ciclo ou um caminho.

Além disso, como não há duas arestas consecutivas de um ciclo de um mesmo emparelhamento (isso implicaria duas arestas de um emparelhamento com mesmo extremo), cada ciclo deve alternar entre arestas de $M - M'$ e de $M' - M$. Implicando assim, em cada ciclo ser par.

Luís Felipe

26/05/26

De volta à volta do Teo de Berge

($\leftarrow$) Seja G um grafo. Se G não contém um caminho M-aumentante, então M é um emparelhamento máximo de G.

Vamos mostrar pela contrapositiva.

Suponha que G tenha um emparelhamento M' de cardinalidade maior do que M. Pelo lema anterior, $F = M \Delta M'$ é constituído por ciclos pares e caminhos. Ciclos pares alternam entre arestas de $M - M'$ e de $M' - M$, portanto possuem mesmo tamanho.

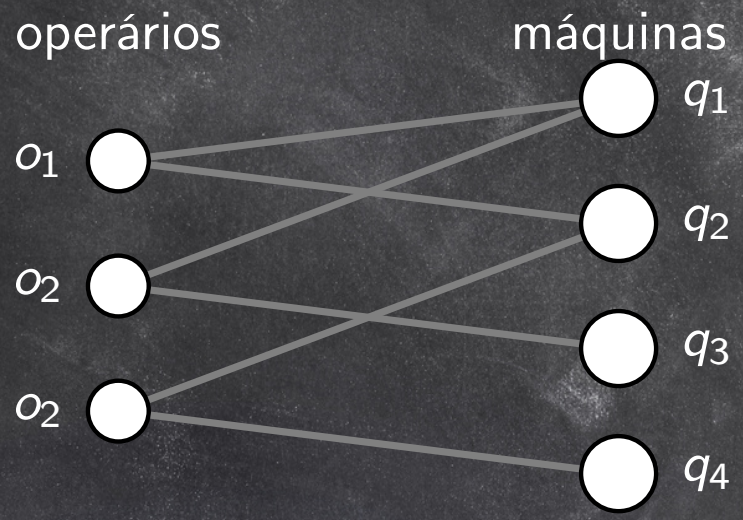
Como $|M'| > |M|$, então F possui, necessariamente, uma componente conexa que é caminho. Além disso, esse caminho deve iniciar e terminar com arestas de $M' - M$, pois $|M'| > |M|$. Assim, este é um caminho M-aumentante.

Luís Felipe

26/05/26

Aplicação 1: alocação operário–máquina

Para cada operário de uma fábrica, há uma lista de máquinas que ele está habilitado a operar. Queremos encontrar uma alocação que maximize o número de pares da forma "operário i opera a máquina j".



Modelagem: construímos um grafo bipartido $G = (O \cup Q, E)$, em que: O é o conjunto de operários, Q é o conjunto de máquinas, e $o_i q_j \in E$ se o_i está habilitado a operar q_j.

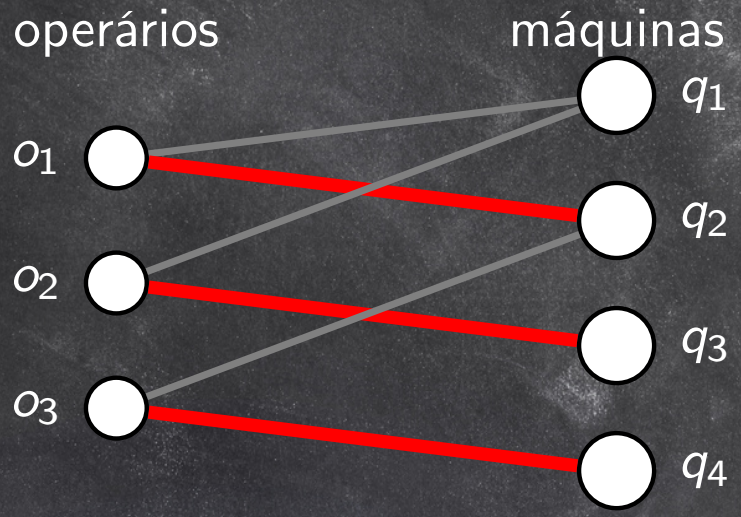
Um emparelhamento máximo em G fornece uma alocação com o maior número possível de pares compatíveis.

Luís Felipe

26/05/26

Aplicação 1: alocação operário–máquina

Para cada operário de uma fábrica, há uma lista de máquinas que ele está habilitado a operar. Queremos encontrar uma alocação que maximize o número de pares da forma "operário i opera a máquina j".



Modelagem: construímos um grafo bipartido $G = (O \cup Q, E)$, em que: O é o conjunto de operários, Q é o conjunto de máquinas, e $o_i q_j \in E$ se o_i está habilitado a operar q_j.

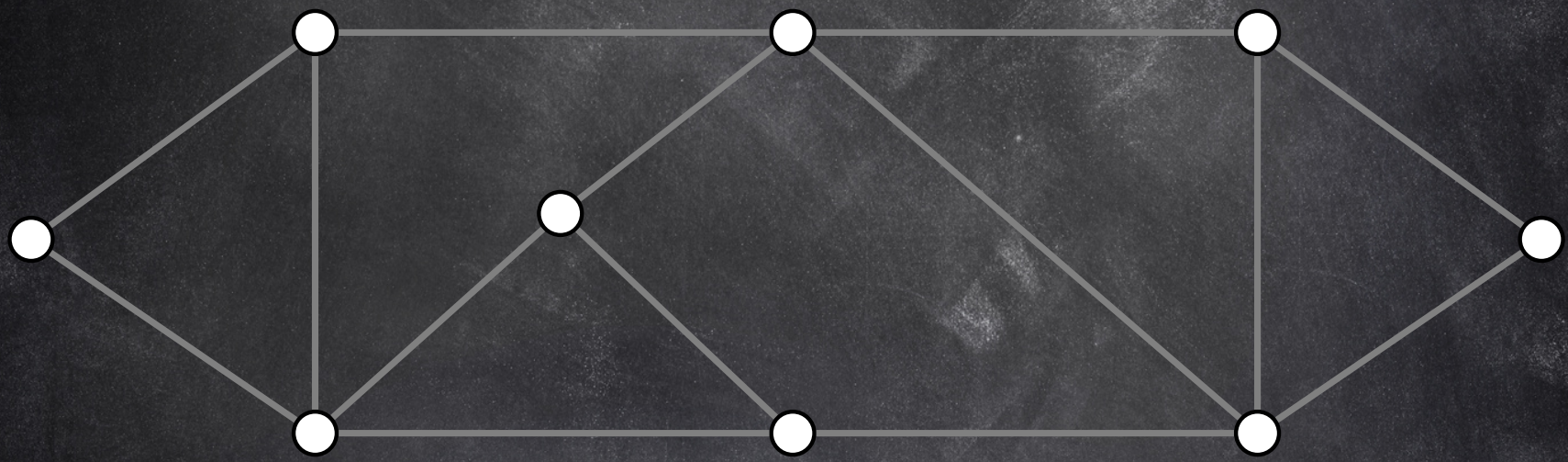
Um emparelhamento máximo em G fornece uma alocação com o maior número possível de pares compatíveis.

Luís Felipe

26/05/26

Aplicação 2: pareamento de processadores

Deseja-se estabelecer pares de processadores parceiros em uma rede. Cada aresta representa dois processadores que podem ser pareados.



O objetivo é escolher o maior número possível de arestas sem extremos em comum, ou seja, um emparelhamento máximo do grafo.

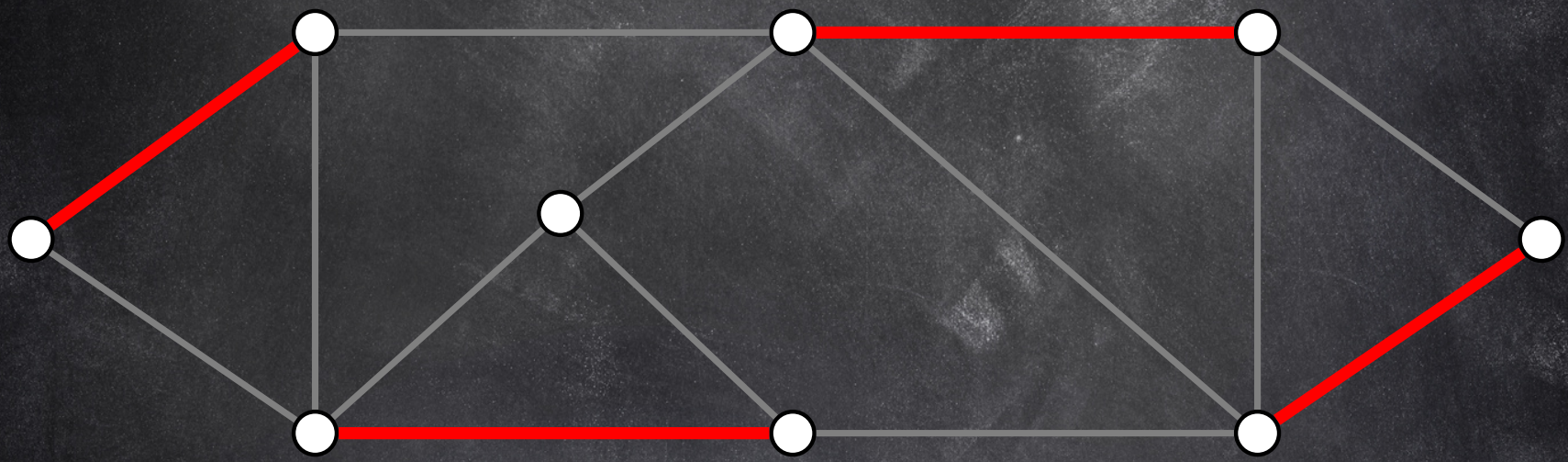
Neste caso, a modelagem não precisa ser bipartida.

Luís Felipe

26/05/26

Aplicação 2: pareamento de processadores

Deseja-se estabelecer pares de processadores parceiros em uma rede. Cada aresta representa dois processadores que podem ser pareados.



O objetivo é escolher o maior número possível de arestas sem extremos em comum, ou seja, um emparelhamento máximo do grafo.

Neste caso, a modelagem não precisa ser bipartida.

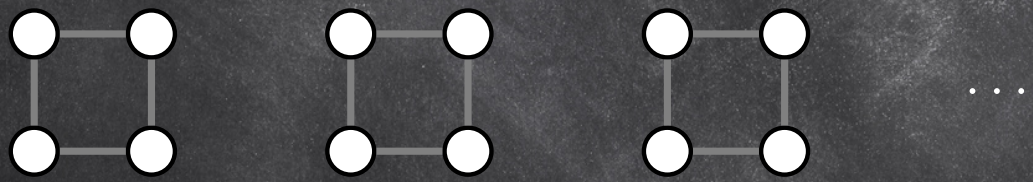
Luís Felipe

26/05/26

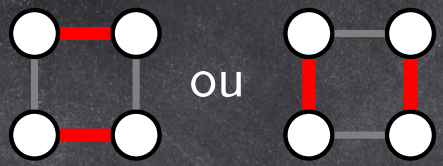
Quantos emparelhamentos máximos podem existir?

Pode haver um número exponencial de emparelhamentos máximos em um grafo.

Considere o grafo formado pela união disjunta de k cópias de C_4.



Em cada cópia de C_4, há exatamente duas escolhas para um emparelhamento perfeito:



Como as escolhas são independentes, o número de emparelhamentos máximos é 2^k. Como $n = 4k$, temos $2^k = 2^{n/4}$.

Luís Felipe

26/05/26

Algoritmo sugerido pelo Teorema de Berge

O Teorema de Berge sugere o seguinte algoritmo: enquanto houver um caminho aumentante, aumente o emparelhamento.

```
1  funcao emparelhamento_maximo(G):
2
3      M <- emparelhamento inicial qualquer
4
5      enquanto existe caminho M-aumentante P em G faca
6
7          M <- M Delta E(P)
8
9      retornar M
```

Aqui, $M \Delta E(P)$ troca, ao longo de P, as arestas que pertencem a M pelas que não pertencem a M.

Quando não existe mais caminho M-aumentante, o Teorema de Berge garante que M é máximo.

Luís Felipe

26/05/26

```python
def norm(e):                          # padroniza a aresta
    return tuple(sorted(e))

def arestas_do_caminho(P):            # arestas consecutivas de P
    A = set()

    for i in range(len(P) - 1):
        A.add(norm((P[i], P[i + 1])))

    return A

def vertices_saturados(M):            # vertices cobertos por M
    S = set()

    for u, v in M:
        S.add(u)
        S.add(v)

    return S
```

Luís Felipe

26/05/26

```python
def dfs_aumentante(G, M, insaturados, origem, v, caminho, anterior):
    # origem: primeiro vertice do caminho
    # v: vertice atual
    # caminho: caminho construido ate agora
    # anterior: None, True ou False
    # None: ainda nao ha aresta anterior
    # True: aresta anterior pertence a M
    # False: aresta anterior nao pertence a M

    if v in insaturados and v != origem:
        return caminho              # caminho alternante entre vertices M-insaturados

    for w in G[v]:                  # w e um vizinho de v

        if w in caminho:
            continue                # evita repetir vertices

        e = norm((v, w))
        atual = e in M              # True se e pertence a M

        if anterior is None or atual != anterior:
            # se e a primeira aresta, ou se alterna com a anterior,
            # continuamos a busca a partir de w

            resp = dfs_aumentante(G, M, insaturados, origem, w, caminho + [w], atual)

            if resp is not None:
                return resp

    return None
```

Luís Felipe

26/05/26

```
50  def busca_caminho_aumentante(G, M):
51
52      M = {norm(e) for e in M}
53
54      saturados = vertices_saturados(M)
55      insaturados = set(G) - saturados
56
57      for s in insaturados:
58
59          P = dfs_aumentante(G, M, insaturados, s, s, [s], None)
60
61          if P is not None:
62              return P
63
64      return None
```

Luís Felipe

26/05/26

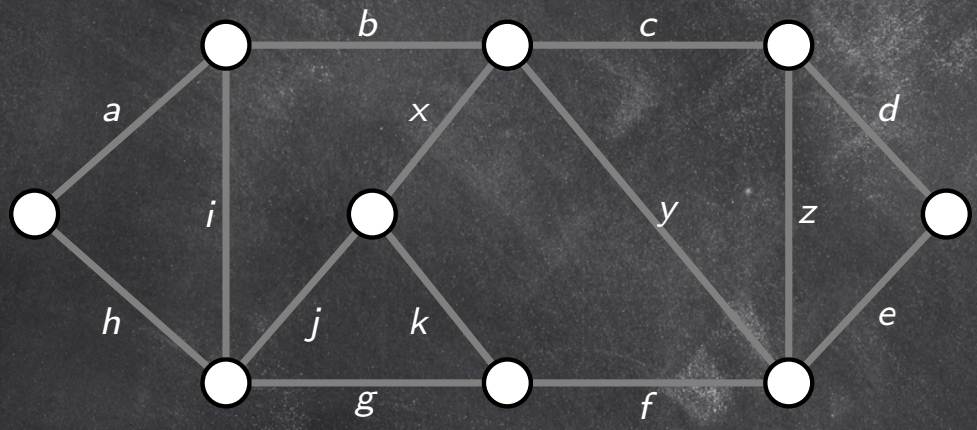
```
65  def emparelhamento_maximo(G, M=None):
66
67      if M is None:
68          M = set()
69      else:
70          M = {norm(e) for e in M}
71
72      while True:
73
74          P = busca_caminho_aumentante(G, M)
75
76          if P is None:
77              return M                # nao ha caminho aumentante
78
79          M = M.symmetric_difference(arestas_do_caminho(P))
```

Observação: esta implementação é útil para simular a ideia do Teorema de Berge. Para obter um algoritmo polinomial em grafos gerais, é preciso uma subrotina eficiente para encontrar caminhos aumentantes, como no algoritmo de Edmonds.

Luís Felipe

26/05/26

Execução em exemplo

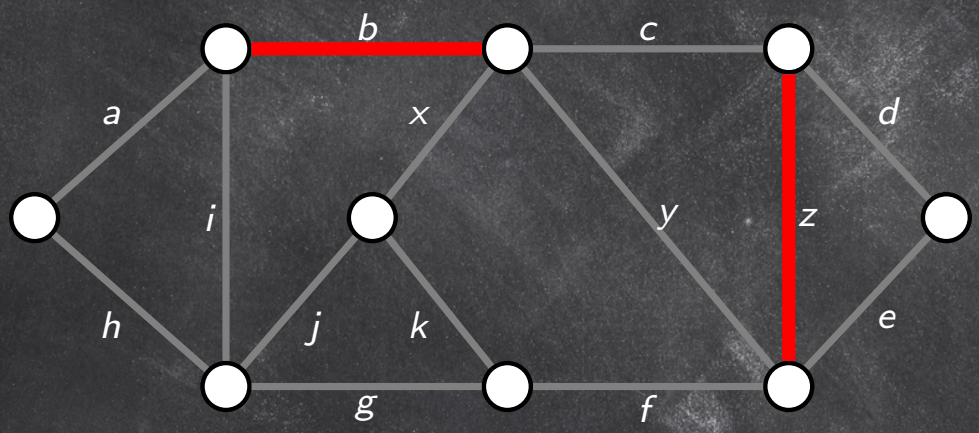


Grafo de entrada. Vamos iniciar com $M = \{b, z\}$.

Luís Felipe

26/05/26

Execução em exemplo

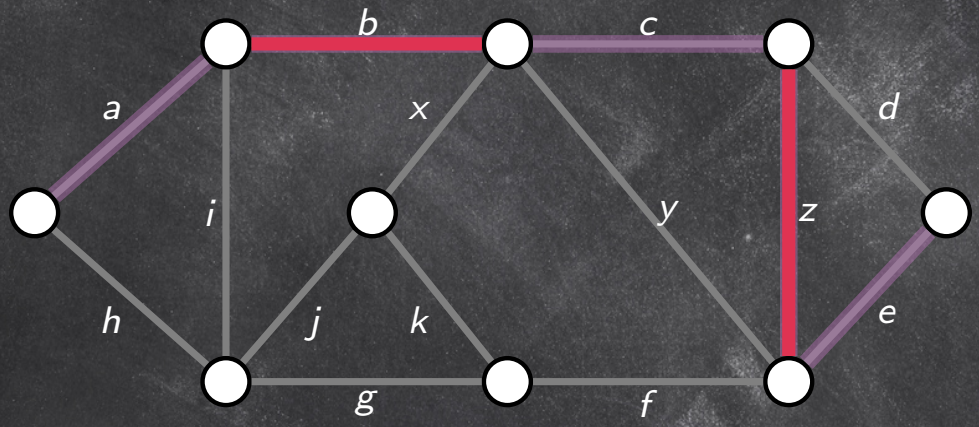


Emparelhamento inicial: $M = \{b, z\}$.

Luís Felipe

26/05/26

Execução em exemplo

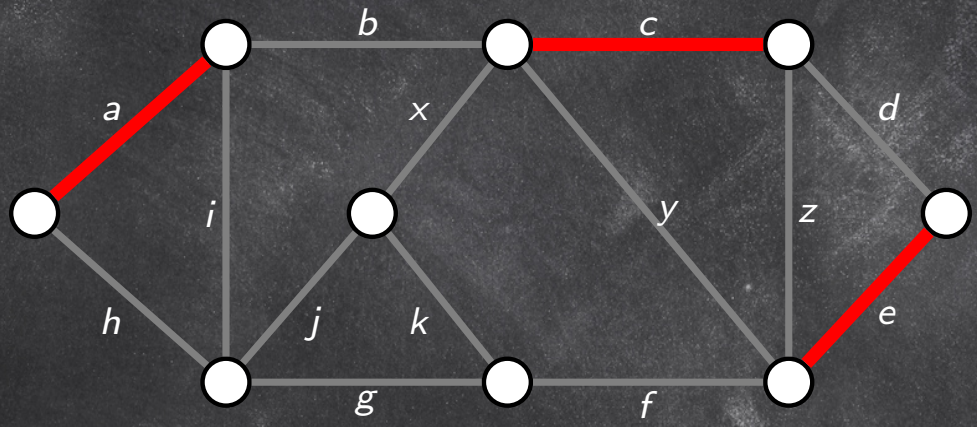


O caminho $P = a, b, c, z, e$ é M-aumentante: começa e termina em vértices M-insaturados e alterna arestas fora e dentro de M.

Luís Felipe

26/05/26

Execução em exemplo



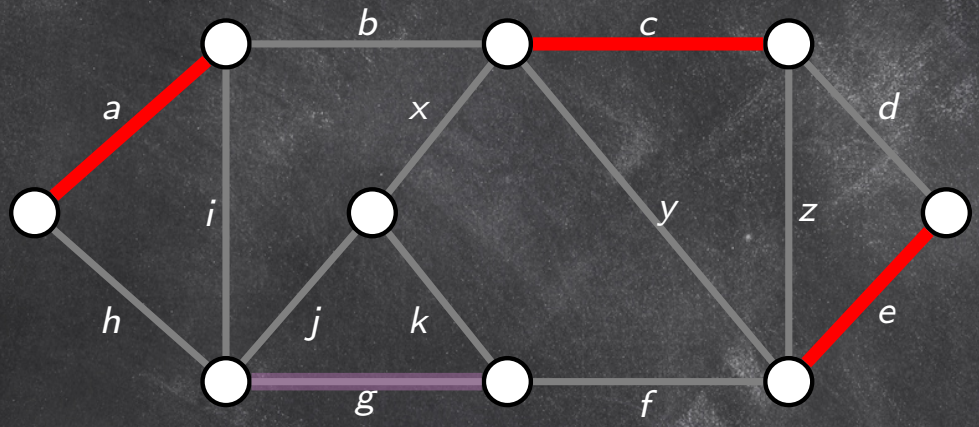
Após a troca ao longo de P:

$$M \leftarrow M \Delta E(P) = \{a, c, e\}.$$

Luís Felipe

26/05/26

Execução em exemplo

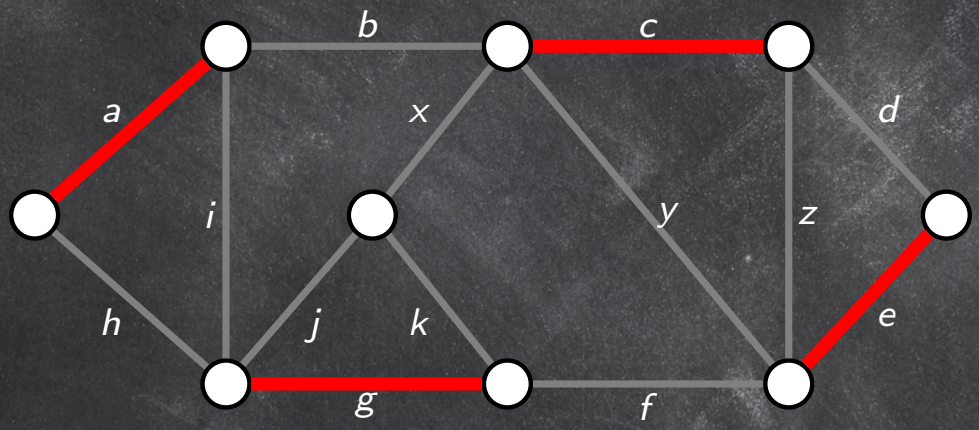


Agora $P = g$ é um caminho M-aumentante de uma única aresta, pois seus dois extremos estão M-insaturados.

Luís Felipe

26/05/26

Execução em exemplo



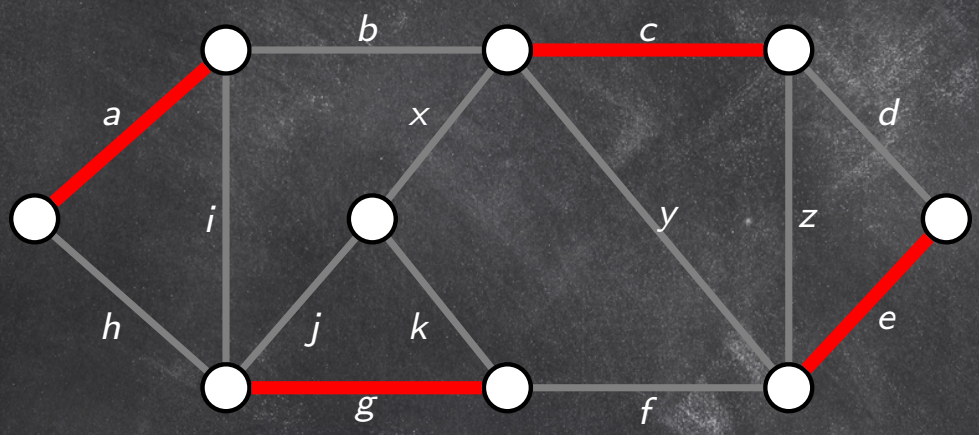
Após a segunda troca:

$$M = \{a, c, e, g\}.$$

Luís Felipe

26/05/26

Execução em exemplo



Não há caminho M-aumentante. Pelo Teorema de Berge, M é máximo.

Luís Felipe

26/05/26

Complexidade do algoritmo de Berge

- Pelo Teorema de Berge, podemos construir um emparelhamento máximo aumentando M enquanto existir caminho M-aumentante.
- A cada iteração, o tamanho de M aumenta em exatamente uma unidade.
- Como $|M| \leq \lfloor n/2 \rfloor$, o algoritmo executa no máximo $\lfloor n/2 \rfloor$ iterações.
- Portanto, se encontrar um caminho M-aumentante custa $T(n, m)$, a complexidade total é:

$$O(n \cdot T(n, m)).$$

- Sem uma subrotina eficiente, podemos procurar um caminho aumentante enumerando todos os caminhos simples do grafo.
- Um grafo com n vértices pode ter $O(n!)$ caminhos simples. Assim, uma busca exaustiva tem custo: $T(n, m) = O(n \cdot n!)$.
- Portanto, o algoritmo de Berge com busca exaustiva tem complexidade: Como $n! \leq n^n$, temos:

$$nn! \leq nn^n = n^{n+1} = 2^{(n+1)\log n}.$$

Logo, $O(nn!) = 2^{O(n \log n)}$.

Luís Felipe

26/05/26

Por que precisamos de Edmonds?

- O Teorema de Berge nos dá uma caracterização muito útil:

$$M \text{ é máximo} \iff \text{não existe caminho } M\text{-aumentante.}$$

- Assim, para encontrar um emparelhamento máximo, basta repetir:

$$M \leftarrow M \Delta E(P),$$

  onde P é um caminho M-aumentante.
- O problema está em encontrar P de forma eficiente.
- Em grafos bipartidos, uma busca alternante é suficiente.
- Em grafos gerais, aparecem ciclos ímpares alternantes, chamados flores.
- O Algoritmo de Edmonds resolve esse problema contraindo flores.

Luís Felipe

26/05/26

Floresta alternante

Seja M um emparelhamento em G.

A busca de Edmonds constrói uma floresta alternante a partir dos vértices M-insaturados.

- As raízes da floresta são vértices M-insaturados.
- Os caminhos da raiz até qualquer vértice alternam arestas fora de M e arestas em M.
- Um vértice é chamado externo se está a distância par da raiz da sua árvore.
- Um vértice é chamado interno se está a distância ímpar da raiz da sua árvore.

raiz = externo, distância par = externo, distância ímpar = interno.

Luís Felipe

26/05/26

Busca alternante: ideia

Durante a busca, tentamos expandir a floresta a partir de vértices externos (distância par).

Seja u um vértice externo e seja uv uma aresta ainda não explorada.

- Se v ainda não está na floresta e v é M-insaturado, encontramos um caminho M-aumentante.
- Se v ainda não está na floresta e v é M-saturado, adicionamos:

$$uv \notin M$$

  e também a aresta de M incidente a v.
- Assim, v entra como vértice interno, e seu par em M entra como vértice externo.
- Se v já é externo, há dois casos importantes.

Luís Felipe

26/05/26

Quando encontramos outro vértice externo

Seja uv uma aresta entre dois vértices externos.

- Se u e v pertencem a árvores diferentes da floresta, então obtemos um caminho M-aumentante.
- Esse caminho é formado por:

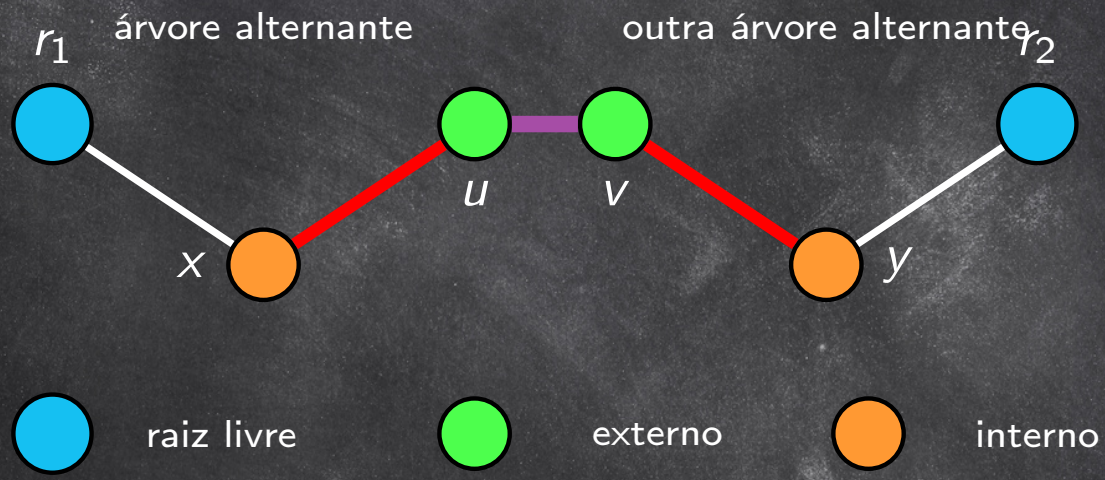
$$\text{raiz de } u \rightsquigarrow u, \quad uv, \quad v \rightsquigarrow \text{raiz de } v.$$

- Como as duas raízes são M-insaturadas, esse é um caminho aumentante.
- Se u e v pertencem à mesma árvore, então a aresta uv fecha um ciclo ímpar alternante.
- Esse ciclo ímpar alternante é uma flor.

Luís Felipe

26/05/26

Aresta entre vértices externos: árvores diferentes



A aresta uv liga dois vértices externos em árvores diferentes.

Assim, obtemos um caminho M-aumentante:

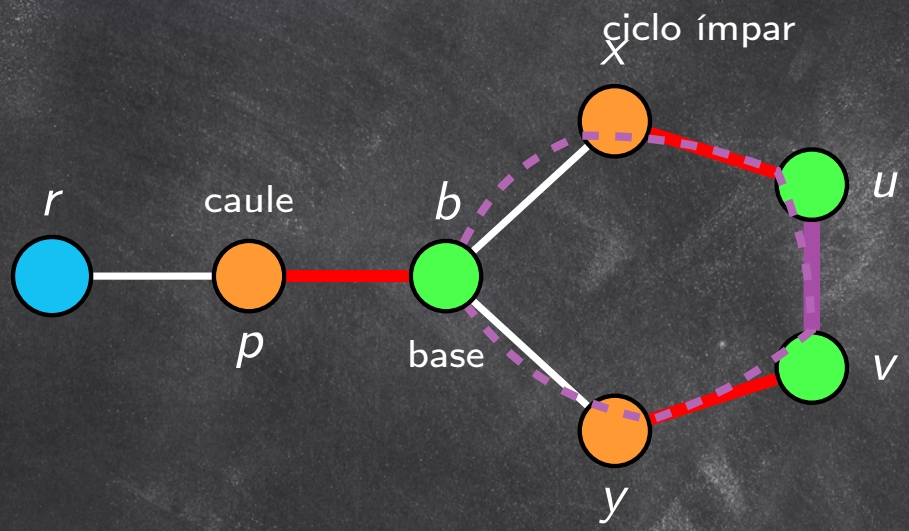
$$r_1 \rightsquigarrow u,\ uv,\ v \rightsquigarrow r_2.$$

As raízes r_1 e r_2 são M-insaturadas.

Luís Felipe

26/05/26

Aresta entre vértices externos: mesma árvore



A aresta uv liga dois vértices externos da mesma árvore.

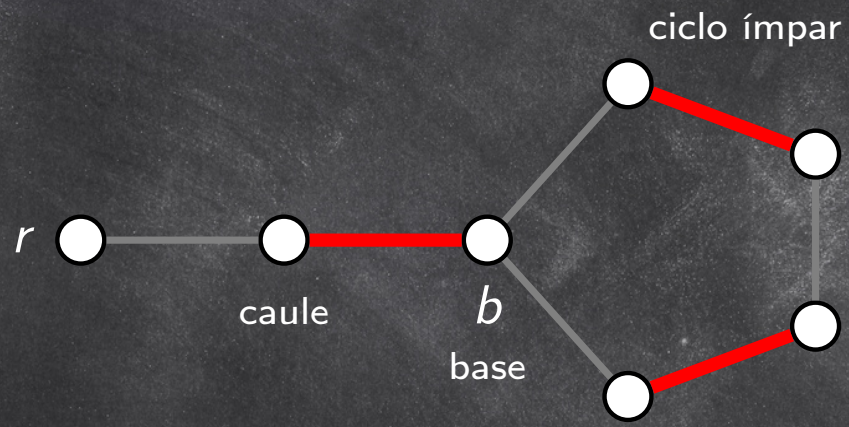
Então o caminho de b até u, a aresta uv, e o caminho de v até b formam um ciclo ímpar.

Esse ciclo é uma flor: ele é quase alternante e tem uma base b.

Luís Felipe

26/05/26

Por que o nome “flor”?



A flor é um ciclo ímpar ligado à árvore alternante por um vértice especial, chamado base.

O caminho da raiz até a base funciona como o “caule”, e o ciclo ímpar é a “flor”.

Neste desenho, a flor por si só não é um caminho aumentante. Ela é uma estrutura que deve ser contraída para que a busca possa continuar.

Luís Felipe

26/05/26

O que é uma flor?

Seja M um emparelhamento em G.

Uma flor é um ciclo ímpar C com um vértice especial b, chamado base, tal que $C - b$ é perfeitamente emparelhado por arestas de M.

Equivalentemente, ao percorrer o ciclo a partir de b, as arestas aparecem na forma:

$$\notin M,\ \in M,\ \notin M,\ \in M,\ \ldots,\ \in M,\ \notin M.$$

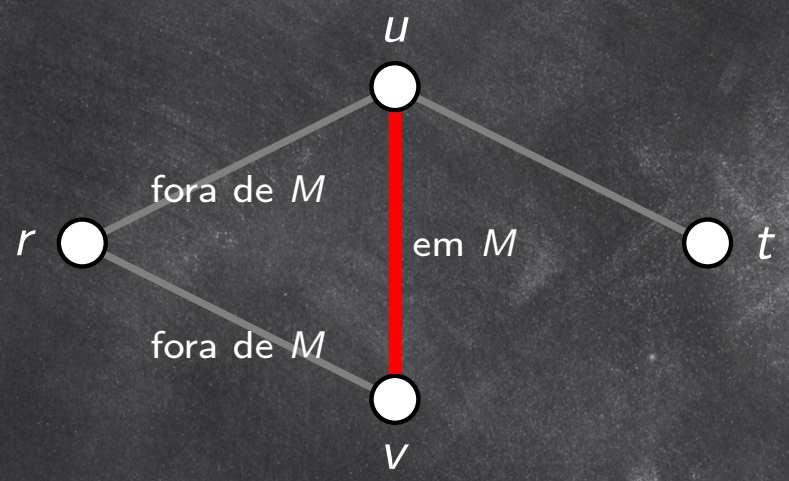
Ou seja, a alternância só falha na base b, onde as duas arestas incidentes a b no ciclo não pertencem a M.

Por isso, uma flor não é um ciclo perfeitamente alternante: isso seria impossível em um ciclo ímpar.

Luís Felipe

26/05/26

Exemplo de flor



O ciclo

$$r, u, v, r$$

é ímpar e alterna arestas fora de M e em M.

Neste exemplo, r é a base da flor.

A aresta uv pertence ao emparelhamento M.

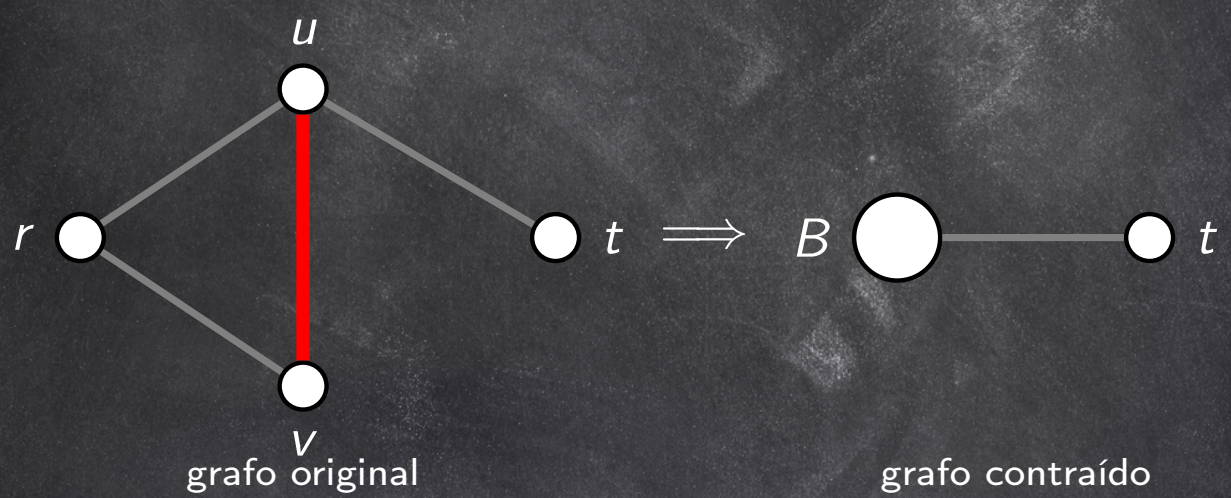
Luís Felipe

26/05/26

Contração da flor

A operação central do algoritmo de Edmonds é a contração.

Contraímos todos os vértices da flor em um único supervértice.



grafo original grafo contraído

A flor passa a ser representada por um único vértice B.

A busca por caminho aumentante continua no grafo contraído.

Luís Felipe

26/05/26

Por que a contração é segura?

A propriedade fundamental é:

G possui caminho M-aumentante $\iff$ G/B possui caminho M/B-aumentante.

Aqui, B é uma flor, e G/B é o grafo obtido contraindo B.

- Se existe caminho aumentante em G que não usa a flor, ele continua existindo em G/B.
- Se um caminho aumentante passa pela flor, no grafo contraído ele passa pelo supervértice B.
- Reciprocamente, se encontramos um caminho aumentante em G/B, podemos expandir B e recuperar um caminho aumentante em G.

Essa é a razão pela qual podemos contrair flores sem perder a existência de caminhos aumentantes.

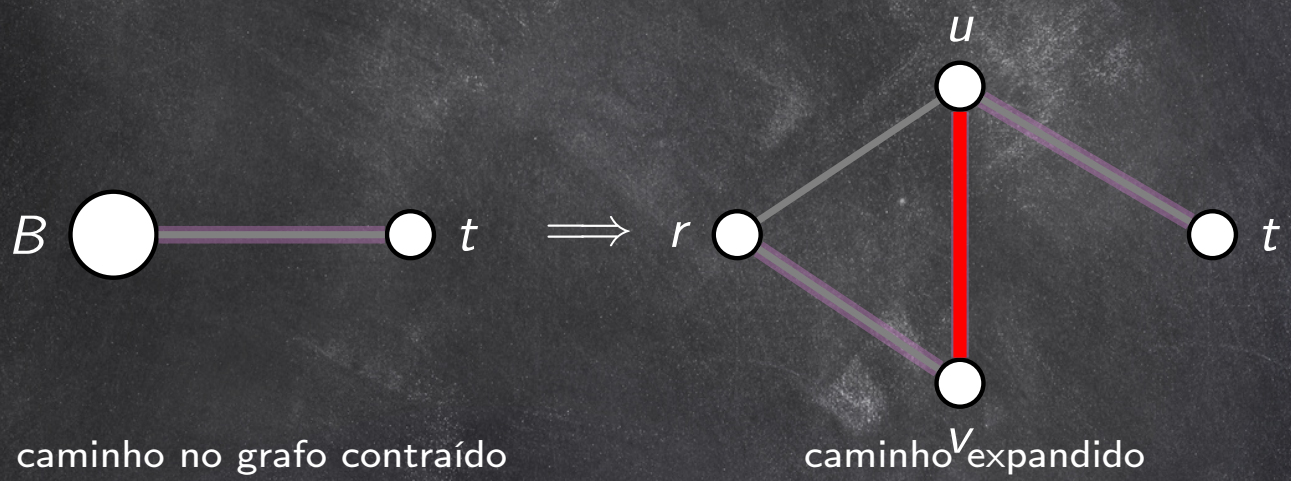
Luís Felipe

26/05/26

Expandindo uma flor

Suponha que encontramos um caminho aumentante no grafo contraído.

Se esse caminho usa o supervértice B, precisamos expandir a flor.



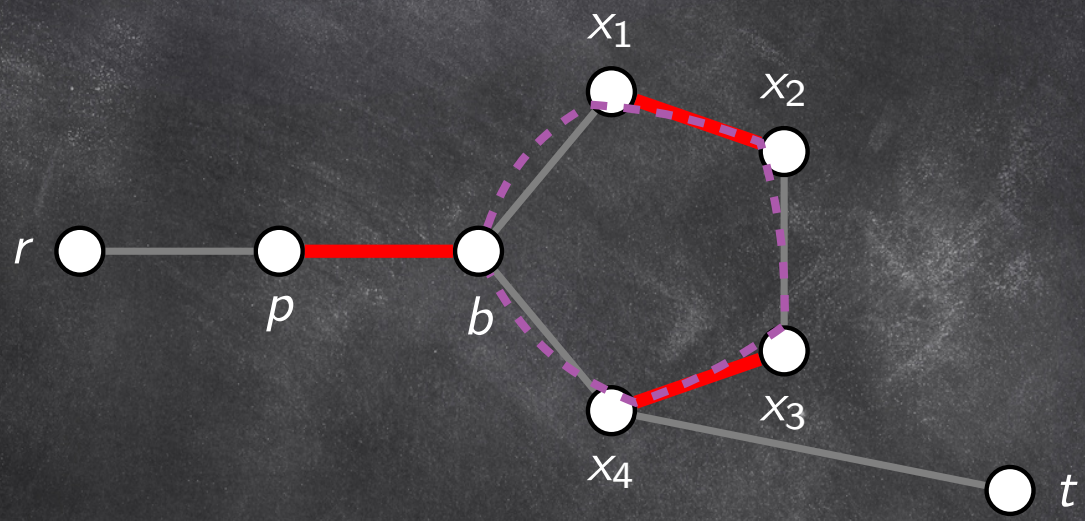
Como a flor é um ciclo ímpar alternante, entre a base e o vértice de saída existe sempre um lado do ciclo que preserva a alternância.

Assim, o caminho aumentante do grafo contraído pode ser levantado para um caminho aumentante no grafo original.

Luís Felipe

26/05/26

Exemplo: contraindo uma flor



Considere o emparelhamento:

$$M = \{pb,\ x_1x_2,\ x_3x_4\}.$$

O ciclo

$$b, x_1, x_2, x_3, x_4, b$$

é uma flor com base b.
Contraímos a flor em um único supervértice B.

Luís Felipe

26/05/26

Exemplo: grafo contraído

No grafo contraído, encontramos o caminho:

$$P' = r, p, B, t.$$

Esse caminho é M-aumentante no grafo contraído:

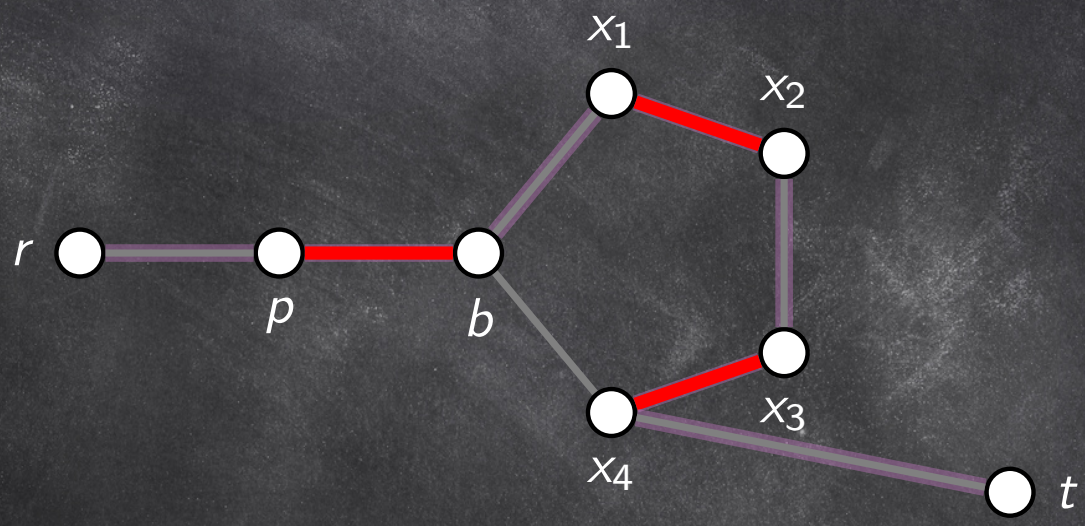
$$rp \notin M, \quad pB \in M, \quad Bt \notin M.$$

Agora precisamos expandir o supervértice B.

Luís Felipe

26/05/26

Exemplo: expandindo a flor



No grafo contraído, o caminho entra na flor pela base b e sai pela aresta correspondente a $x_4 t$.
Para expandir, escolhemos dentro da flor o caminho alternante de b até x_4:

$$b, x_1, x_2, x_3, x_4.$$

Assim, obtemos no grafo original:

$$P = r, p, b, x_1, x_2, x_3, x_4, t.$$

Luís Felipe

26/05/26

Exemplo: caminho aumentado no grafo original

O caminho expandido é:

$$P = r, p, b, x_1, x_2, x_3, x_4, t.$$

Observe a alternância:

$$rp \notin M, \quad pb \in M, \quad bx_1 \notin M, \quad x_1x_2 \in M, \quad x_2x_3 \notin M, \quad x_3x_4 \in M, \quad x_4t \notin M.$$

Logo, P é um caminho M-aumentante.

Agora podemos fazer:

$$M' = M \Delta E(P).$$

As arestas de M no caminho são removidas, e as arestas fora de M no caminho são adicionadas.

Luís Felipe

26/05/26

Exemplo: após o aumento

Antes:

$$M = \{pb,\ x_1x_2,\ x_3x_4\}.$$

Caminho aumentante:

$$P = r, p, b, x_1, x_2, x_3, x_4, t.$$

Depois:

$$M' = M\Delta E(P).$$

Portanto:

$$M' = \{rp,\ bx_1,\ x_2x_3,\ x_4t\}.$$

O tamanho do emparelhamento aumentou em uma unidade:

$$|M'| = |M| + 1.$$

Luís Felipe

26/05/26

Busca aumentante com flores

```
1   funcao busca_edmonds(G, M):
2
3       inicialize uma floresta alternante F
4       raizes de F sao os vertices M-insaturados
5
6       enquanto existir vertice externo u nao processado faca
7
8           para cada aresta uv incidente a u faca
9
10              se v nao esta em F entao
11
12                  se v eh M-insaturado entao
13                      retornar caminho aumentante encontrado
14                  senao
15                      adicione uv a F
16                      adicione a aresta de M incidente a v
17                      marque v como interno
18                      marque o par de v em M como externo
19
20              senao se v e externo entao
21
22                  se u e v estao em arvores diferentes entao
23                      retornar caminho aumentante encontrado
24                  senao
25                      encontre a flor formada por uv
26                      contraia a flor
27                      continue a busca no grafo contraido
28
29      retornar vazio
```

Luís Felipe

26/05/26

Algoritmo de Edmonds

```
1  funcao edmonds(G):
2
3      M <- conjunto vazio
4
5      enquanto verdadeiro faca
6
7          P <- busca_edmonds(G, M)
8
9          se P = vazio entao
10             retornar M
11         fim-se
12
13         M <- M Delta E(P)
14
15     fim-enquanto
```

Ideia: a função busca_edmonds procura um caminho M-aumentante. Quando encontra uma flor, contrai a flor e continua a busca.

Se um caminho aumentante é encontrado no grafo contraído, ele é expandido para o grafo original.

Luís Felipe

26/05/26

Resumo dos casos da busca

Durante a busca, processamos arestas que saem de vértices externos.

Caso encontrado	Ação
v não está na floresta e é insaturado	achamos caminho aumentante
v não está na floresta e é saturado	expandimos a floresta
v é externo em outra árvore	achamos caminho aumentante
v é externo na mesma árvore	encontramos uma flor
v é interno	ignoramos a aresta

O único caso que não aparece em grafos bipartidos é o caso da flor.

Esse é exatamente o caso tratado por Edmonds.

Luís Felipe

26/05/26

Complexidade do algoritmo de Edmonds

- O algoritmo de Edmonds segue a estratégia de Berge:

$$M \leftarrow M \Delta E(P),$$

  onde P é um caminho M-aumentante.
- Cada aumento incrementa $|M|$ em exatamente uma unidade.
- Como $|M| \leq \lfloor n/2 \rfloor$, há no máximo $O(n)$ aumentos.
- A diferença está na subrotina que encontra P: em grafos gerais, Edmonds usa contração de flores.
- Em uma busca aumentante, cada contração reduz o número de vértices do grafo contraído.
- Logo, há no máximo $O(n)$ contrações durante uma busca.
- Entre contrações, podemos examinar as arestas do grafo, o que custa $O(m)$.
- Portanto, uma busca aumentante custa $O(nm)$.
- Como há no máximo $O(n)$ aumentos, a complexidade total é:

$$O(n) \cdot O(nm) = O(n^2 m).$$