# Aula 20 - Algoritmos Aproximativos

Luís Felipe

UFF

05 de Dezembro de 2023

Luís Felipe
05/12/23

# O que são?

- Dentre os algoritmos que não garantem a corretude da resposta, temos os Algoritmos Aproximativos
- Agora, contudo, consideraremos os problemas de otimização
- Imaginem que temos um problema NP-c ou um problema em P com complexidade ruim ou, ainda, um problema que não se conheça a complexidade
- Algoritmos Aproximativos nem sempre dão a resposta correta, mas a solução dada é próxima da solução real do problema com melhora no tempo de execução.
  - Quão próxima? Depende do fator de aproximação.
  - Como assegurar esse fator dado que não se conhece a solução real?
    - Geralmente, esse fator é calculado com base em certas condições necessárias ou suficientes para a solução.
    - Tais propriedades conduzem a limites que nos permitem calcular o fator de aproximação.

Luís Felipe
05/12/23

# Minimização vs maximização

- Minimização $\frac{A}{c(otima)} \leq \alpha, \alpha \geq 1$
- Maximização $\frac{A}{c(otima)} \geq \alpha, 0 < \alpha \leq 1$
  - $\alpha$ é o fator de aproximação

Luís Felipe
05/12/23

# Melhor aprox. vs Melhor tempo

- Imaginem que tenhamos dois algoritmos para um mesmo problema de otimização.
- O primeiro tem razão 1.5, dito 1.5-aproximativo, e complexidade $O(n \log n)$
- O segundo tem razão 1.375, dito 1.375-aproximativo, e complexidade $O(n^2)$
- Sendo assim:
  - Com que tipo de problema de otimização estamos lidando? Minimização
  - Qual o melhor algoritmo aproximativo? Depende

Luís Felipe
05/12/23

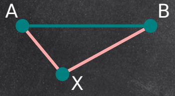
# Aproximativos vs heurísticas

- Qual a diferença entre um algoritmo aproximativo e uma heurística?
  - A heurística não fornece um fator de aproximação
- Mas então por que temos heurísticas?
  - Porque nem sempre temos as tais condições que conduzem a limites que permitem calcular o fator de aproximação

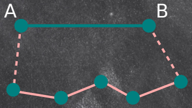
Luís Felipe
05/12/23

# Caixeiro-Viajante Métrico (Euclidiano)

1. Grafo Completo
2. Os custos das arestas satisfazem à desigualdade triangular




$c(A,B) \leq c(A,X) + c(X,B)$




3. Quer-se determinar o ciclo Hamiltoniano de menor custo

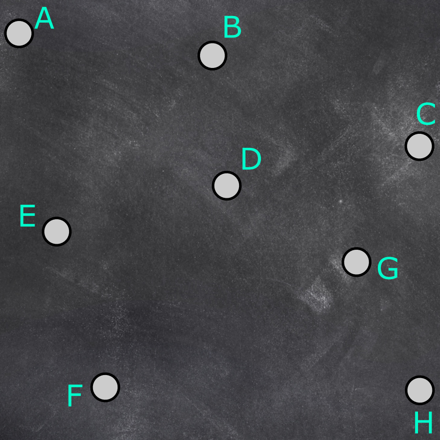
Luís Felipe
05/12/23

# Algoritmo RSL – Rosenkrantz, Stearns e Lewis

- Obtém-se uma árvore geradora mínima
- Duplicam-se as arestas da árvore, obtendo um multigrafo.
  - Consequência: todo vértice tem grau par
  - Logo, o grafo resultante é Euleriano
- Toma-se um circuito Euleriano
  - A partir dele, constrói-se um ciclo Hamiltoniano substituindo as repetições de vértices por caminhos "diretos"
  - Assim, arestas que não estão no circuito Euleriano serão inseridas.
  - Note que elas sempre existem, pois o grafo original é completo.

Luís Felipe
05/12/23

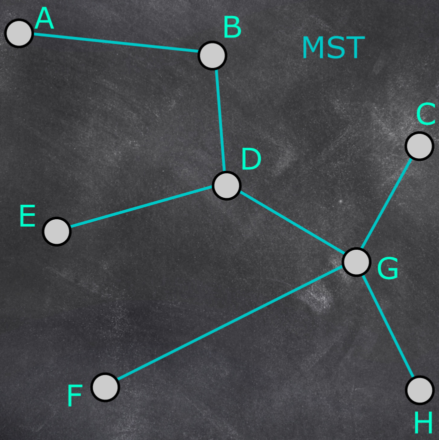
# Exemplo de execução – RSL



Luís Felipe

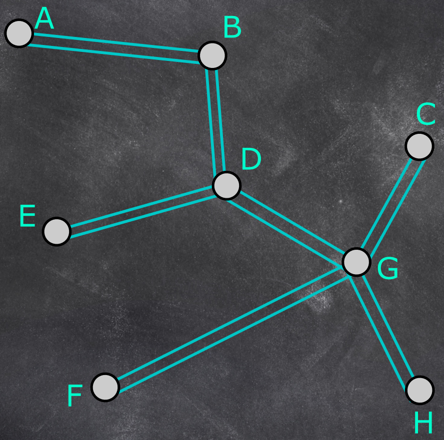
05/12/23

# Exemplo de execução – RSL



Luís Felipe
05/12/23

# Exemplo de execução – RSL

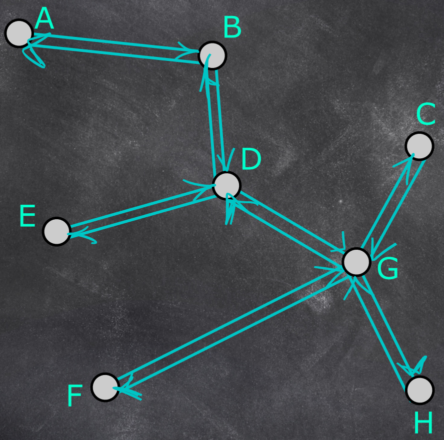



Duplicam-se as arestas da MST

Luís Felipe

05/12/23

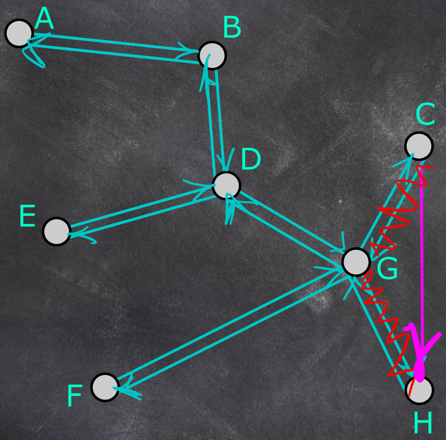
# Exemplo de execução – RSL




Obtem um circuito Euleriano:
A, B, D, G, C, G, H, G, F, G, D, E, D, B, A

Luís Felipe
05/12/23

# Exemplo de execução – RSL



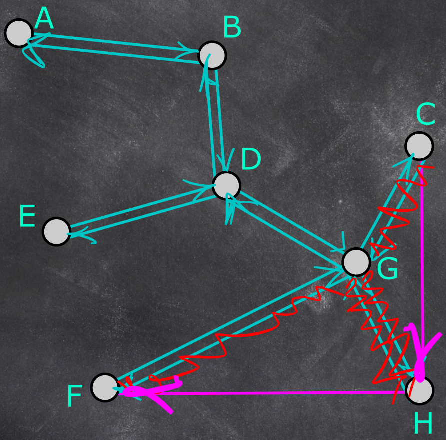

Obtém ciclo Hamiltoniano

**A**, **B**, **D**, **G**, **C**, G, **H**, G, F, G, D, E, D, B, A

Luís Felipe

05/12/23

# Exemplo de execução – RSL



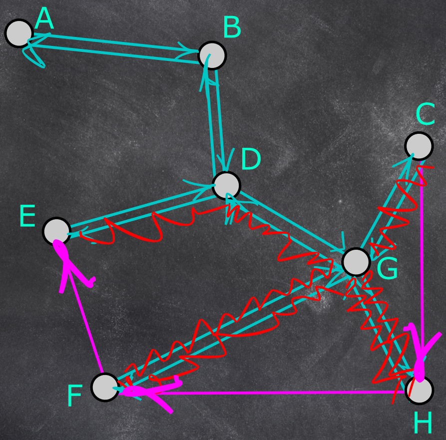

Obtém ciclo Hamiltoniano

**A**, **B**, **D**, **G**, **C**, G, **H**, G, **F**, G, D, E, D, B, A

Luís Felipe

05/12/23

# Exemplo de execução – RSL



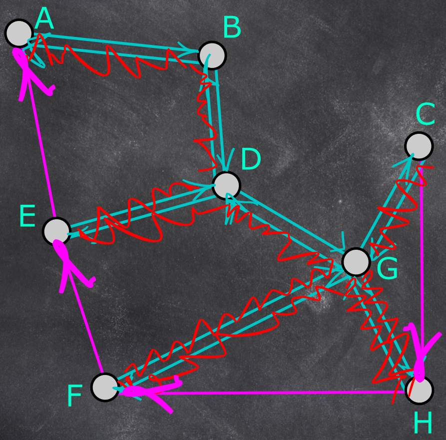

Obtém ciclo Hamiltoniano

**A**, **B**, **D**, **G**, **C**, G, **H**, G, **F**, G, D, **E**, D, B, A

Luís Felipe

05/12/23

# Exemplo de execução – RSL

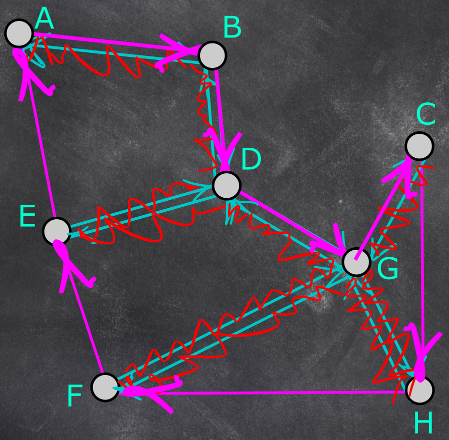



Obtém ciclo Hamiltoniano

**A**, **B**, **D**, **G**, **C**, G, **H**, G, **F**, G, D, **E**, D, B, **A**

Luís Felipe
05/12/23

# Exemplo de execução – RSL



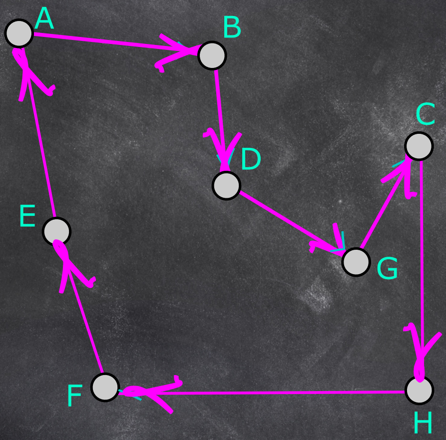

Obtém ciclo Hamiltoniano

**A**, **B**, **D**, **G**, **C**, G, **H**, G, **F**, G, D, **E**, D, B, **A**

Luís Felipe

05/12/23

# Exemplo de execução




Obtém ciclo Hamiltoniano

**A**, **B**, **D**, **G**, **C**, **H**, **F**, **E**, **A**

Luís Felipe
05/12/23

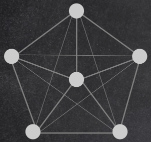
# Fator de aproximação e complexidade

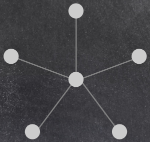
- Sejam:
  $c(T)$ o custo da árvore geradora mínima,
  $c(\text{ótimo})$ o custo ótimo do Caixeiro Viajante e
  $c(A)$ o custo do algoritmo RSL
  - $c(T) \leq c(\text{ótimo})$
  - Note que $c(A) \leq 2c(T)$ pela desigualdade triangular. Basicamente, $c(A)$ na primeira etapa de $A$ é igual a $2c(T)$, pela obtenção do ciclo Hamiltoniano do multigrafo na cópia das arestas de $T$, depois reduzimos $c(A)$ pelo "corte de caminho" garantido da desigualdade triangular na adição de novas arestas
  - Como $2c(T) \leq 2c(\text{ótimo})$, temos $c(A) \leq 2c(\text{ótimo})$
  - O algoritmo RSL é 2-aproximativo
  - Complexidade: $O(n^2)$ após ter obtido a MST.

Luís Felipe
05/12/23

# Sobre o fator do RSL

- Seria possível melhorar o fator de aproximação do RSL?
- Veremos agora uma instância para a qual a resposta fornecida pelo RSL é, assintoticamente, 2 vezes o valor ótimo
- O custo das arestas em negrito é 1 e o das demais é 2





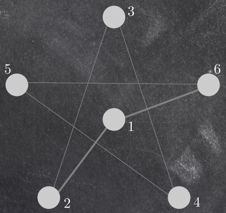
- Note que o CICLO Hamiltoniano de menor custo tem custo *n*.

Luís Felipe

05/12/23

# Qual o custo?

- Suponha que a MST obtida seja a estrela $K_{1,5}$ e que o circuito Euleriano seja o seguinte:




- Note que, neste caso, o custo da solução do algoritmo é $\frac{4n}{2} - 2 = 2n - 2$, que, assintoticamente é $2n$.

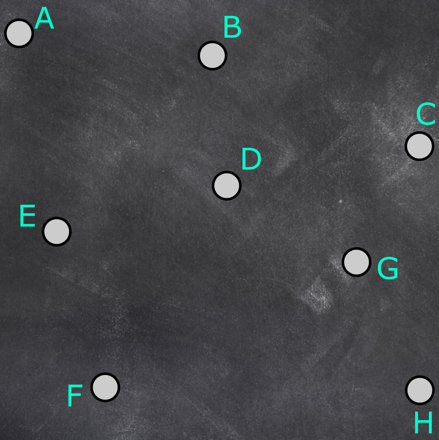
Luís Felipe


# Algoritmo de Christofides

- Obtém-se uma árvore geradora mínima
- Toma-se um emparelhamento perfeito mínimo no grafo sobre o conjunto dos vértices de grau ímpar e adicionam-se tais arestas à arvore
    - ▶ Sempre temos um emparelhamento perfeito? Sim, pois o grafo é completo e o número de vértices de grau ímpar em qualquer grafo é par (Teoria dos Grafos)
    - ▶ Consequência da adição de arestas: Grafo Euleriano
- Toma-se um circuito Euleriano
    - ▶ A partir dele, constrói-se um ciclo Hamiltoniano substituindo as repetições de vértices por caminhos "diretos"
    - ▶ Assim, arestas que não estão no circuito Euleriano serão inseridas.
    - ▶ Note que elas sempre existem, pois o grafo original é completo.

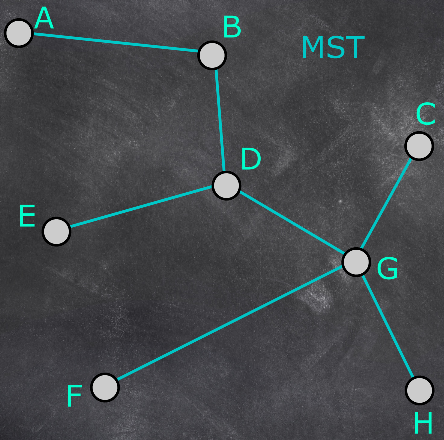
Luís Felipe
05/12/23

# Exemplo de execução – Christofides



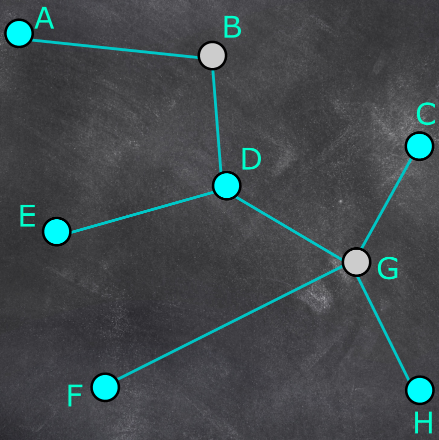
Luís Felipe
05/12/23

# Exemplo de execução – Christofides



Luís Felipe
05/12/23

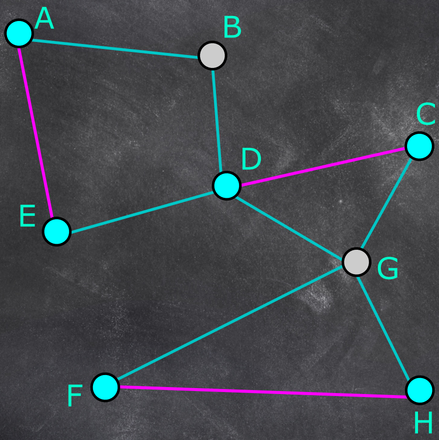
# Exemplo de execução – Christofides



Luís Felipe

05/12/23

# Exemplo de execução – Christofides

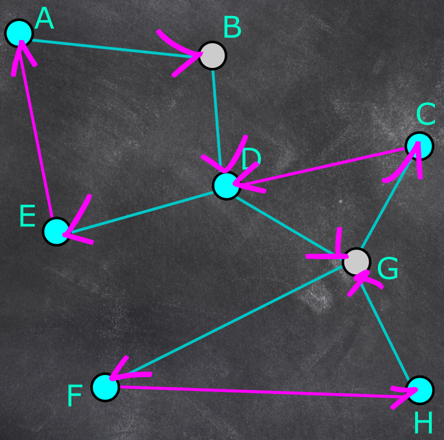



Emparelhamento perfeito de custo mínimo
sobre o subgrafo dos vértices de grau ímpar

Luís Felipe

05/12/23

# Exemplo de execução – Christofides

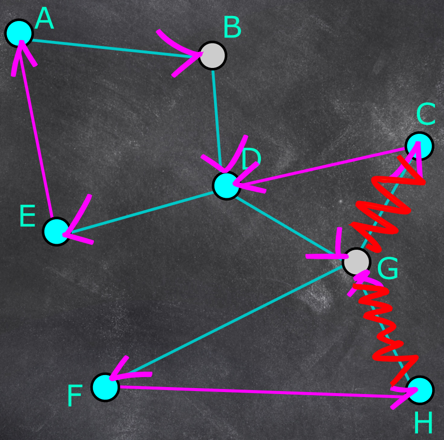



Circuito Euleriano de todo o grafo
A, B, D, G, F, H, G, C, D, E, A

Luís Felipe

05/12/23

# Exemplo de execução – Christofides

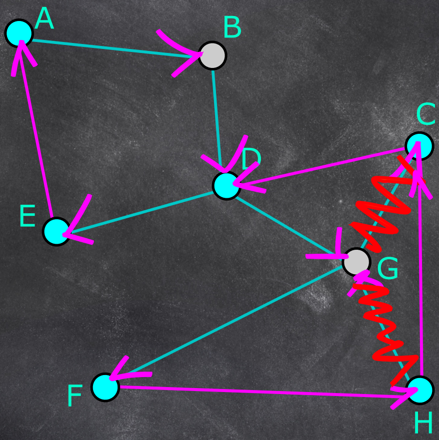



Ciclo Hamiltoniano resultante

**A**, **B**, **D**, **G**, **F**, **H**, G, C, D, E, A

Luís Felipe
05/12/23

# Exemplo de execução - Christofides



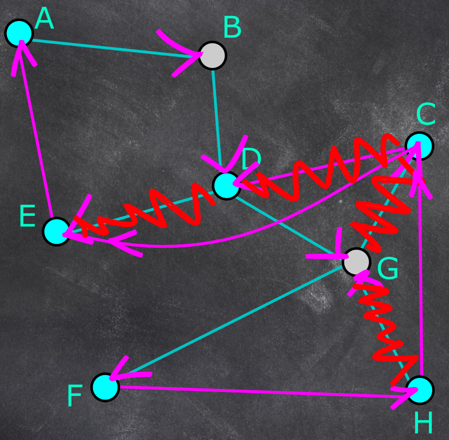

Ciclo Hamiltoniano resultante

**A**, **B**, **D**, **G**, **F**, **H**, G, **C**, D, E, A

Luís Felipe

05/12/23

# Exemplo de execução – Christofides

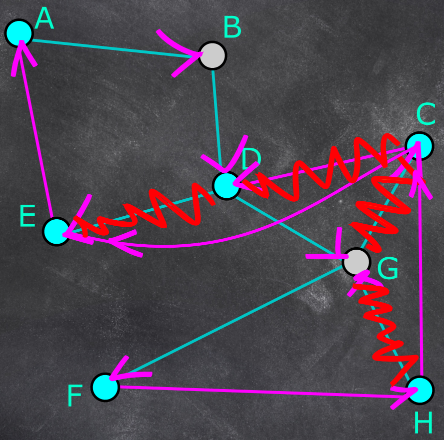



Ciclo Hamiltoniano resultante

**A**, **B**, **D**, **G**, **F**, **H**, G, **C**, D, **E**, A

Luís Felipe
05/12/23

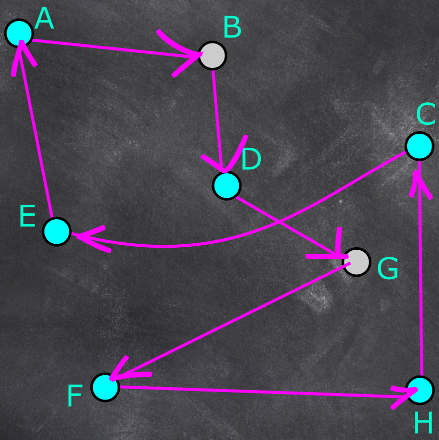
# Exemplo de execução - Christofides




Ciclo Hamiltoniano resultante
**A**, **B**, **D**, **G**, **F**, **H**, G, **C**, D, **E**, **A**

Luís Felipe
05/12/23

# Exemplo de execução – Christofides




Ciclo Hamiltoniano resultante
**A, B, D, G, F, H, C, E, A**

Luís Felipe
05/12/23

# Fator de aproximação e complexidade

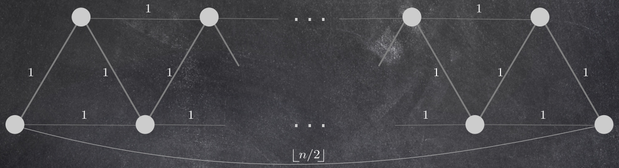
- Sejam:
  $c(T)$ o custo da árvore geradora mínima,
  $c(M)$ o custo do emparelhamento perfeito,
  $c(\text{ótimo})$ o custo ótimo e
  $c(A)$ o custo do algoritmo de Christofides
  - $c(T) \leq c(\text{ótimo})$
  - $c(M) \leq \frac{c(\text{ótimo})}{2}$, pois usamos no máximo metade das arestas do custo ótimo no emparelhamento, dado que usamos no máximo $\frac{|V|}{2}$ arestas e no ciclo temos $|V|$ arestas
  - $c(A) \leq c(T) + c(M)$, pela desigualdade triangular e pelo fato de usarmos arestas de $T$ e de $M$
  - Logo, $c(A) \leq c(\text{ótimo}) + \frac{c(\text{ótimo})}{2} = \frac{3}{2}c(\text{ótimo})$
  - O algoritmo de Christofides é 1.5-aproximativo
  - Complexidade: $O(n^3)$ pelo algoritmo para obter emparelhamento perfeito mínimo (Algoritmos em Grafos)

Luís Felipe

05/12/23

# Sobre o fator do Christofides

- Será que é possível melhorar o fator de aproximação do Christofides?
- Veremos agora uma instância para a qual a resposta fornecida pelo Christofides é exatamente 1.5 vezes o valor ótimo
- O custo da aresta curva é $\lfloor \frac{n}{2} \rfloor$ e o das demais é 1. O grafo tem número ímpar de vértices.

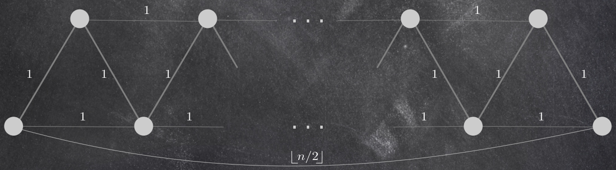



- Note que o CICLO Hamiltoniano de menor custo tem custo $n$.

Luís Felipe
05/12/23

# Qual o custo?

- Suponha que a MST obtida seja induzida pelas arestas em negrito e que o circuito Euleriano seja esse caminho + a aresta curva.




- Note que, neste caso, o custo da solução do algoritmo é $(n-1) + \lfloor \frac{n}{2} \rfloor$.

Luís Felipe
05/12/23

# Vertex Cover

- Vertex Cover: Cobertura das arestas por vértices.
- Já vimos que a versão de otimização do Vertex Cover é NP-difícil.
- Seria possível obtermos um algoritmo aproximativo para o VC?
- Sabemos que o Problema do Emparelhamento Máximo é solucionável em tempo polinomial.
- Seria este problema um aliado para uma algoritmo aproximativo para VC?

Luís Felipe
05/12/23

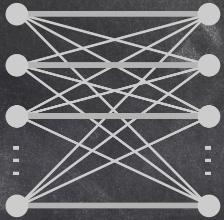
# Emparelhamento vs VC

- $c(otimo)$: tamanho do menor vertex cover.
- Seja $M$ um emparelhamento máximo
- $|M| \leq c(otimo)$ cada aresta de $M$ precisa ser coberta por um vértice
- Como $M$ é maximal, os vértices que são extremos das arestas de $M$ cobrem todas as arestas
  - Caso contrário, existiria pelo menos uma aresta para adicionarmos a $M$. Contradição com a maximalidade de $M$
  - A solução dada pelo algoritmo é, portanto, o conjunto de todos os vértices incidentes a $M$.
- A quantidade de vértices que são incidentes às arestas de $M$ é $2|M|$. Seja $A$ a quantidade $2|M|$ de vértices retornada pelo algoritmo.
- Daí, $2|M| \leq 2c(otimo)$. Assim: $\frac{A}{c(otimo)} = \frac{2|M|}{c(otimo)} \leq \frac{2c(otimo)}{c(otimo)}$. Logo, este algoritmo é 2-aproximativo.

Luís Felipe
05/12/23

# Sobre o fator de aproximação

- Seria possível melhorar o fator de aproximação deste algoritmo aproximativo para VC?
- Não. Observe o exemplo: A figura descreve uma família infinita de grafos $K_{n,n}$ onde o emparelhamento máximo é perfeito, com cardinalidade $2n$, enquanto que o Vertex Cover mínimo tem cardinalidade $n$.



Luís Felipe
05/12/23

# Sobre o fator de aproximação (II)

- Será que então conseguimos subir o limite inferior?
  - Dada a abordagem com emparelhamento, não.
  - Considerando $K_n$, com $n$ ímpar, o emparelhamento máximo tem cardinalidade $\frac{n-1}{2}$, enquanto que o vertex cover mínimo tem tamanho $(n-1)$.