Seminários de Combinatória do IME
Universidade Federal Fluminense

Dificuldade e Eficiência de Problemas em Grafos, Strings e Permutações

Luís Felipe Ignácio Cunha

21/08/2019

Atividades em pesquisa

Combinatória, Complexidade computacional e Algoritmos (exatos e aproximativos)

Atividades em pesquisa

Combinatória, Complexidade computacional e Algoritmos (exatos e aproximativos)

- Grafos:
  - Admissibilidade
  - Tesselabilidade
  - Convexidade
- Strings:
  - Compressão
  - Blocos comuns
  - Mediana
- Permutações:
  - Ordenação
  - Diâmetro
  - Centralidade

- Uma árvore geradora T de um grafo G é t-geradora se cada par de vértices adjacentes em G possui distância no máximo t em T

- Uma árvore geradora T de um grafo G é t-geradora se cada par de vértices adjacentes em G possui distância no máximo t em T
- G é chamado de grafo t-admissível

- Uma árvore geradora T de um grafo G é t-geradora se cada par de vértices adjacentes em G possui distância no máximo t em T
- G é chamado de grafo t-admissível
- O índice de extensão de G, denotado por $\sigma_T(G)$, é o menor t tal que G seja t-admissível

- Uma árvore geradora T de um grafo G é t-geradora se cada par de vértices adjacentes em G possui distância no máximo t em T
- G é chamado de grafo t-admissível
- O índice de extensão de G, denotado por $\sigma_T(G)$, é o menor t tal que G seja t-admissível

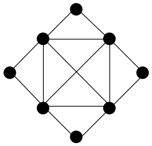
Problema de admissibilidade
Entrada: Um grafo conexo G e um inteiro t.
Pergunta: $\sigma_T(G) \leq t$?

Exemplo

- Objetivo: determinar o menor t tal que exista uma árvore cuja distância entre vértices adjacentes do grafo seja no máximo t.

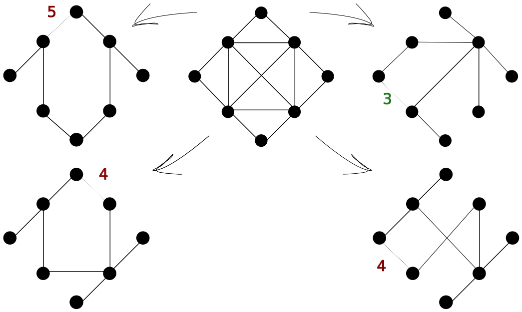
Exemplo

- Objetivo: determinar o menor t tal que exista uma árvore cuja distância entre vértices adjacentes do grafo seja no máximo t.



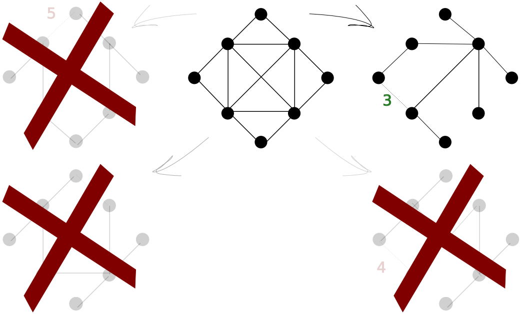
Exemplo

- Objetivo: determinar o menor t tal que exista uma árvore cuja distância entre vértices adjacentes do grafo seja no máximo t.



Exemplo

- Objetivo: determinar o menor t tal que exista uma árvore cuja distância entre vértices adjacentes do grafo seja no máximo t.



$$\sigma_T(G) \leq 3$$

Complexidade

- Determinar se $\sigma_T(G) \leq 2$ é solucionável em tempo polinomial.

Complexidade

- Determinar se $\sigma_T(G) \leq 2$ é solucionável em tempo polinomial.
- Determinar se $\sigma_T(G) \leq t, t \geq 4$ é NP-completo

Complexidade

- Determinar se $\sigma_T(G) \leq 2$ é solucionável em tempo polinomial.
- Determinar se $\sigma_T(G) \leq t, t \geq 4$ é NP-completo
- Porém, determinar se $\sigma_T(G) = 3$ permanece em aberto

Complexidade

- Determinar se $\sigma_T(G) \leq 2$ é solucionável em tempo polinomial.
- Determinar se $\sigma_T(G) \leq t, t \geq 4$ é NP-completo
- Porém, determinar se $\sigma_T(G) = 3$ permanece em aberto
  - L. Cai, D. G. Corneil, *Tree spanners*, SIAM J. Discrete Math, 1995.

Complexidade

- Determinar se $\sigma_T(G) \leq 2$ é solucionável em tempo polinomial.
- Determinar se $\sigma_T(G) \leq t, t \geq 4$ é NP-completo
- Porém, determinar se $\sigma_T(G) = 3$ permanece em aberto
  - L. Cai, D. G. Corneil, *Tree spanners*, SIAM J. Discrete Math, 1995.

Portanto, abordagens interessantes são:

1. determinar grafos 3-admissíveis
2. determinar a complexidade do problema para classes de grafos

Resultados

Resultados

- Determinamos o índice de extensão dos grafos com poucos P_4's
  - generalizamos resultados conhecidos dos grafos livres de P_4.

Resultados

- Determinamos o índice de extensão dos grafos com poucos P_4's
  - generalizamos resultados conhecidos dos grafos livres de P_4.
- Determinamos parcialmente a dicotomia P vs NP-completo do problema para os grafos (k, ℓ)
  - grafos (k, ℓ) são grafos cujo conjunto de vértices pode ser particionado em k conjuntos independentes e ℓ cliques.
  - generalizamos resultados conhecidos para alguns valores de k e ℓ.

Resultados

- Determinamos o índice de extensão dos grafos com poucos P_4's
  - generalizamos resultados conhecidos dos grafos livres de P_4.
- Determinamos parcialmente a dicotomia P vs NP-completo do problema para os grafos (k, ℓ)
  - grafos (k, ℓ) são grafos cujo conjunto de vértices pode ser particionado em k conjuntos independentes e ℓ cliques.
  - generalizamos resultados conhecidos para alguns valores de k e ℓ.
- Determinamos o índice de extensão para subclasses de classes cujo problema é NP-completo
  - grafos cordal $(2, 1)$.

Estado da arte

Sabíamos que...

$k \backslash \ell$	0	1	2	$\ldots$
0	–	P	?	$\ldots$
1	–	P	?	$\ldots$
2	NP-c	?	?	$\ldots$
3	?	?	?	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Estado da arte
Sabemos que...

k \ ℓ	0	1	2	3	$\dots$	$f(n)$	$\dots$
0	–	P	P	?	$\dots$	NP-c	$\dots$
1	–	P	?	?	$\dots$	NP-c	$\dots$
2	NP-c	NP-c	NP-c	NP-c	NP-c	NP-c	$\dots$
3	NP-c	NP-c	NP-c	NP-c	NP-c	NP-c	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Células cinza são nossos resultados.

Desafios

Desafios:

- Estudar isomorfismo de árvores para prevenção de árvores repetidas em estratégias força-bruta;
- Determinar a caracterização dos grafos 3-admissíveis (questão em aberto há mais de 20 anos).

Publicações
Admissibilidade

1. F. Couto, L. Cunha.
   Theoretical Computer Science.
   Submetido ao Special Issue do COCOA'18 para convidados. 2019.

2. F. Couto, L. Cunha.
   Electronic Notes in Theoretical Computer Science.
   X Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS'19).

3. F. Couto, L. Cunha, D. Juventude.
   LI Simpósio Brasileiro de Pesquisa Operacional (SBPO'19).

4. F. Couto, L. Cunha.
   Lecture Notes in Computer Science.
   12th Annual International Conference on Combinatorial Optimization and Applications (COCOA'18).

5. F. Couto, L. Cunha, D. Ferraz.
   VIII Latin American Workshop on Cliques in Graphs (LAWCG'18).

Colaborações
Admissibilidade

- Universidade Federal Rural do Rio de Janeiro:
  - profa. Fernanda Couto
  - Daniel Juventude (aluno de IC)
  - Diego Ferraz (aluno de TCC)

Tesselabilidade

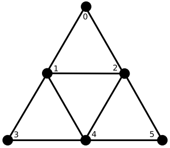
- Modelos de caminhadas quântica em redes são desenvolvidos com o intuito de melhorar a eficiência computacional
- Modelos escalonados motivam tesselações em grafos

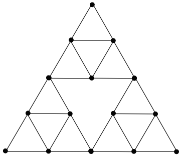
Tesselabilidade

- Modelos de caminhadas quântica em redes são desenvolvidos com o intuito de melhorar a eficiência computacional
- Modelos escalonados motivam tesselações em grafos
- Uma tesselação é uma partição dos vértices em cliques.
  - Objetivo: determinar o menor t tal que existam t tesselações que cubram as arestas do grafo.

Tesselabilidade

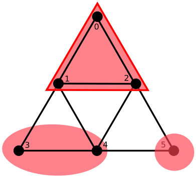
- Modelos de caminhadas quântica em redes são desenvolvidos com o intuito de melhorar a eficiência computacional
- Modelos escalonados motivam tesselações em grafos
- Uma tesselação é uma partição dos vértices em cliques.
  - Objetivo: determinar o menor t tal que existam t tesselações que cubram as arestas do grafo.

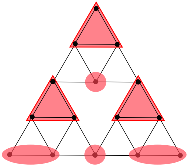




Tesselabilidade

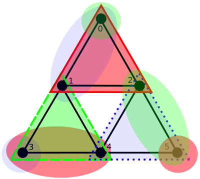
- Modelos de caminhadas quântica em redes são desenvolvidos com o intuito de melhorar a eficiência computacional
- Modelos escalonados motivam tesselações em grafos
- Uma tesselação é uma partição dos vértices em cliques.
  - Objetivo: determinar o menor t tal que existam t tesselações que cubram as arestas do grafo.

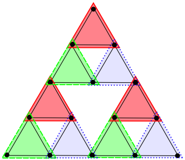




Tesselabilidade

- Modelos de caminhadas quântica em redes são desenvolvidos com o intuito de melhorar a eficiência computacional
- Modelos escalonados motivam tesselações em grafos
- Uma tesselação é uma partição dos vértices em cliques.
  - Objetivo: determinar o menor t tal que existam t tesselações que cubram as arestas do grafo.





Complexidade

- Determinar se $t \leq 2$ é solucionável em tempo polinomial.
  - R. Portugal et al., The staggered quantum walk model, Quantum Inf Process, 2016.

Complexidade

- Determinar se $t \leq 2$ é solucionável em tempo polinomial.
  - R. Portugal et al., The staggered quantum walk model, Quantum Inf Process, 2016.

Questões interessantes são:

1. Caracterizar grafos t-tesseláveis
2. Relacionar com outros problemas em grafos

Relações com problemas de coloração

- $t \leq \min\{\chi'(G), \chi(K(G))\}$.

Relações com problemas de coloração

- $t \leq \min\{\chi'(G), \chi(K(G))\}$.
- Se G é livre de K_3, então $t = \chi'(G)$.

Relações com problemas de coloração

- $t \leq \min\{\chi'(G), \chi(K(G))\}$.
- Se G é livre de K_3, então $t = \chi'(G)$.
  - é NP-completo decidir se um grafo livre de K_3 é 3-tesselável;

Relações com problemas de coloração

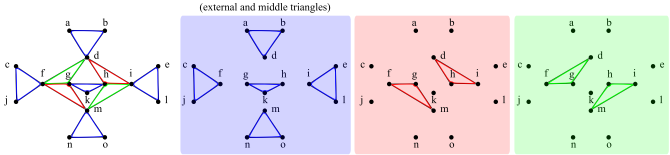
- $t \leq \min\{\chi'(G), \chi(K(G))\}$.
- Se G é livre de K_3, então $t = \chi'(G)$.
  - é NP-completo decidir se um grafo livre de K_3 é 3-tesselável;
  - o problema é solucionável em tempo polinomial para grafos bipartidos.

Relações com problemas de coloração

- $t \leq \min\{\chi'(G), \chi(K(G))\}$.
- Se G é livre de K_3, então $t = \chi'(G)$.
  - é NP-completo decidir se um grafo livre de K_3 é 3-tesselável;
  - o problema é solucionável em tempo polinomial para grafos bipartidos.
- Decidir se um grafo planar com $\Delta \leq 6$ é 3-tesselável é NP-completo

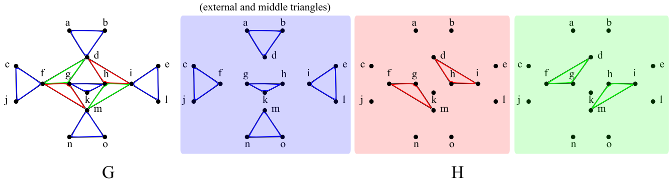
Relações com problemas de coloração

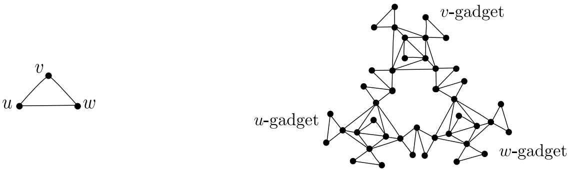
- $t \leq \min\{\chi'(G), \chi(K(G))\}$.
- Se G é livre de K_3, então $t = \chi'(G)$.
  - é NP-completo decidir se um grafo livre de K_3 é 3-tesselável;
  - o problema é solucionável em tempo polinomial para grafos bipartidos.
- Decidir se um grafo planar com $\Delta \leq 6$ é 3-tesselável é NP-completo



Relações com problemas de coloração

- $t \leq \min\{\chi'(G), \chi(K(G))\}$.
- Se G é livre de K_3, então $t = \chi'(G)$.
  - é NP-completo decidir se um grafo livre de K_3 é 3-tesselável;
  - o problema é solucionável em tempo polinomial para grafos bipartidos.
- Decidir se um grafo planar com $\Delta \leq 6$ é 3-tesselável é NP-completo





Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

- t-tesselabilidade para qualquer $t \geq 4$
- 4-tesselabilidade para grafos $(2, 1)$
- 3-tesselabilidade para grafos livres de diamantes com diâmetro 5

Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

- t-tesselabilidade para qualquer $t \geq 4$
- 4-tesselabilidade para grafos $(2, 1)$
- 3-tesselabilidade para grafos livres de diamantes com diâmetro 5

Obtivemos algoritmo de melhor complexidade para o reconhecimento de grafos 2-tesseláveis.

Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

- t-tesselabilidade para qualquer $t \geq 4$
- 4-tesselabilidade para grafos $(2, 1)$
- 3-tesselabilidade para grafos livres de diamantes com diâmetro 5

Obtivemos algoritmo de melhor complexidade para o reconhecimento de grafos 2-tesseláveis.

Desafios:

Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

- t-tesselabilidade para qualquer $t \geq 4$
- 4-tesselabilidade para grafos $(2, 1)$
- 3-tesselabilidade para grafos livres de diamantes com diâmetro 5

Obtivemos algoritmo de melhor complexidade para o reconhecimento de grafos 2-tesseláveis.

Desafios:

- Tratar classes que possuam alto valor de diâmetro.

Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

- t-tesselabilidade para qualquer $t \geq 4$
- 4-tesselabilidade para grafos $(2, 1)$
- 3-tesselabilidade para grafos livres de diamantes com diâmetro 5

Obtivemos algoritmo de melhor complexidade para o reconhecimento de grafos 2-tesseláveis.

Desafios:

- Tratar classes que possuam alto valor de diâmetro.
- Como tamanho da maior estrela induzida é um limite inferior, buscamos caracterizar grafos perfeitamente tesseláveis

Caracterizações

Além disso, desenvolvemos outras provas de NP-completudes para:

- t-tesselabilidade para qualquer $t \geq 4$
- 4-tesselabilidade para grafos $(2, 1)$
- 3-tesselabilidade para grafos livres de diamantes com diâmetro 5

Obtivemos algoritmo de melhor complexidade para o reconhecimento de grafos 2-tesseláveis.

Desafios:

- Tratar classes que possuam alto valor de diâmetro.
- Como tamanho da maior estrela induzida é um limite inferior, buscamos caracterizar grafos perfeitamente tesseláveis
  - Assim como o tamanho da maior clique se relaciona com os grafos perfeitos.

Publicações
Tesselabilidade

1. A. Abreu, L. Cunha, C. de Figueiredo, L. Kowada, F. Marquezino, D. Posner, R. Portugal.
   Theoretical Computer Science.
   Aceito, *minor reviews*. 2019.

2. A. Abreu, L. Cunha, C. de Figueiredo, L. Kowada, F. Marquezino, D. Posner, R. Portugal.
   Submetido ao Theoretical Computer Science. 2019.

3. A. Abreu, L. Cunha, T. Fernandes, C. de Figueiredo, L. Kowada, F. Marquezino, D. Posner, R. Portugal.
   Lecture Notes in Computer Science.
   13th Latin American Symposium on Theoretical Informatics (LATIN'18).

4. A. Abreu, L. Cunha, T. Fernandes, C. de Figueiredo, L. Kowada, F. Marquezino, D. Posner, R. Portugal.
   Matemática Contemporânea, 2017.

5. A. Abreu, L. Cunha, L. Kowada, F. Marquezino.
   VII Latin American Workshop on Cliques in Graphs (LAWCG'16).

Colaborações
Tesselabilidade

- Universidade Federal Fluminense:
  - prof. Luis Kowada

Colaborações

- Universidade Federal Fluminense:
  - prof. Luis Kowada
- Laboratório Nacional de Computação Científica:
  - prof. Renato Portugal

- Universidade Federal Fluminense:
  - prof. Luis Kowada
- Laboratório Nacional de Computação Científica:
  - prof. Renato Portugal
- Universidade Federal do Rio de Janeiro:
  - profa. Celina de Figueiredo
  - prof. Franklin Marquezino
  - Alenxandre Santiago (aluno de doutorado)
  - Daniel Posner (pós-doc)

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um **bloco maximal haplótipo** é uma tripla (K, i, j):

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita),

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	0	1	1	0	0
1	0	0	0	0	1	1	0
1	1	1	0	1	0	1	0

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	**0**	**1**	1	0	0
1	0	0	0	0	1	1	0
1	1	1	**0**	**1**	0	1	0

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	**0**	1	1	0	0
1	0	0	**0**	0	1	1	0
1	1	1	**0**	1	0	1	0

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	0	1	1	0	0
1	0	0	0	0	1	1	0
1	1	1	0	1	0	1	0

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \dots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \dots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \dots j] = s_q[i \dots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \dots j] = s_q[i \dots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	0	1	1	0	0
1	0	0	0	0	1	**1**	**0**
1	1	1	0	1	0	**1**	**0**

- Blocos haplótipos: Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	0	1	1	0	**0**
1	0	0	0	0	1	1	**0**
1	1	1	0	1	0	1	**0**

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \dots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \dots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \dots j] = s_q[i \dots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \dots j] = s_q[i \dots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	0	1	1	0	0
1	0	0	0	0	1	1	0
1	1	1	0	1	0	1	0

Existem alguns outros blocos haplótipos.

- **Blocos haplótipos:** Obter de um conjunto de k strings binárias de tamanho n todos os blocos maximais de elementos comuns.

Entrada: $S = (s_1, \ldots, s_k)$, cada uma de tamanho n.

Um bloco maximal haplótipo é uma tripla (K, i, j):

$K \subseteq \{1, \ldots, k\}$, $|K| \geq 2$, e $1 \leq i \leq j \leq n$, tal que:

1. $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S|_K$,
2. $i = 1$ ou $s_p[i-1] \neq s_q[i-1]$, para $s_p, s_q \in S$ (maximal-esquerda),
3. $j = n$ ou $s_p[j+1] \neq s_q[j+1]$, para $s_p, s_q \in S$ (maximal-direita), e
4. $\nexists K'$, tal que $K \subset K'$ e $s_p[i \ldots j] = s_q[i \ldots j]$, para todos $s_p, s_q \in S$.

Exemplo:

0	1	0	0	1	1	0	0
1	0	0	0	0	1	1	0
1	1	1	0	1	0	1	0

Existem alguns outros blocos haplótipos. Existem $O(kn)$ blocos haplótipos maximais em S.

Resultados e Desafios

- Obtivemos algoritmo com complexidade de tempo $O(kn^2)$.

Resultados e Desafios

- Obtivemos algoritmo com complexidade de tempo $O(kn^2)$.

Desafios:

1. Como a saída possui tamanho $O(kn)$, buscamos por uma estratégia melhor do que $O(kn^2)$.

Resultados e Desafios

- Obtivemos algoritmo com complexidade de tempo $O(kn^2)$.

Desafios:

1. Como a saída possui tamanho $O(kn)$, buscamos por uma estratégia melhor do que $O(kn^2)$.

Resultados e Desafios

- Obtivemos algoritmo com complexidade de tempo $O(kn^2)$.

Desafios:

1. Como a saída possui tamanho $O(kn)$, buscamos por uma estratégia melhor do que $O(kn^2)$.

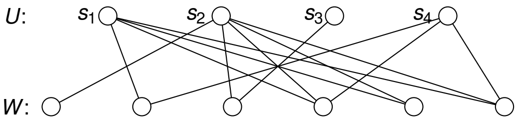


Figura: Grafo bipartido $(U \cup W, E)$

$$
\begin{array}{lcccccc}
s_1 = & 0 & 1 & 0 & 1 & 1 & 1 \\
s_2 = & 1 & 0 & 1 & 1 & 1 & 1 \\
s_3 = & 0 & 0 & 1 & 0 & 0 & 0 \\
s_4 = & 0 & 1 & 0 & 1 & 0 & 1
\end{array}
$$

- Soluções são os gêmeos em U em relação a intervalos consecutivos em W.

Resultados e Desafios

- Obtivemos algoritmo com complexidade de tempo $O(kn^2)$.

Desafios:

1. Como a saída possui tamanho $O(kn)$, buscamos por uma estratégia melhor do que $O(kn^2)$.

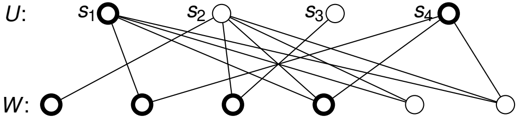


Figura: Grafo bipartido $(U \cup W, E)$

$$
\begin{array}{lcccccc}
s_1 = & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & 1 & 1 \\
s_2 = & 1 & 0 & 1 & 1 & 1 & 1 \\
s_3 = & 0 & 0 & 1 & 0 & 0 & 0 \\
s_4 = & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & 0 & 1
\end{array}
$$

- Soluções são os gêmeos em U em relação a intervalos consecutivos em W.

Resultados e Desafios

- Obtivemos algoritmo com complexidade de tempo $O(kn^2)$.

Desafios:

1. Como a saída possui tamanho $O(kn)$, buscamos por uma estratégia melhor do que $O(kn^2)$.

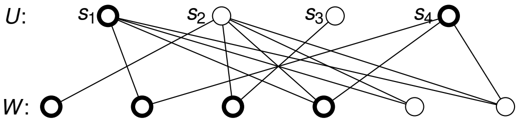


Figura: Grafo bipartido $(U \cup W, E)$

$$
\begin{array}{lcccccc}
s_1 = & 0 & 1 & 0 & 1 & 1 & 1 \\
s_2 = & 1 & 0 & 1 & 1 & 1 & 1 \\
s_3 = & 0 & 0 & 1 & 0 & 0 & 0 \\
s_4 = & 0 & 1 & 0 & 1 & 0 & 1
\end{array}
$$

- Soluções são os gêmeos em U em relação a intervalos consecutivos em W.
- Todos os gêmeos podem ser determinados em tempo $O(n^2(|U| + |W| + |E|))$.

Indexação de padrões em strings

Entrada: string $s \in \{0, 1\}^n$.

Consulta: existe uma substring de tamanho $k \leq n$ com $i \leq k$ cópias de 1s?

Indexação de padrões em strings

Entrada: string $s \in \{0,1\}^n$.

Consulta: existe uma substring de tamanho $k \leq n$ com $i \leq k$ cópias de 1s?

Resultados:

- Construímos a tabela de índices com o certificado em tempo $O(n)$ utilizando $O(n)$-palavras de espaço.

Indexação de padrões em strings

Entrada: string $s \in \{0, 1\}^n$.

Consulta: existe uma substring de tamanho $k \leq n$ com $i \leq k$ cópias de 1s?

Resultados:

- Construímos a tabela de índices com o certificado em tempo $O(n)$ utilizando $O(n)$-palavras de espaço.
- Construímos a tabela de índices em tempo $O(n)$ utilizando $O(n)$-bit de espaço.

Indexação de padrões em strings

Entrada: string $s \in \{0, 1\}^n$.

Consulta: existe uma substring de tamanho $k \leq n$ com $i \leq k$ cópias de 1s?

Resultados:

- Construímos a tabela de índices com o certificado em tempo $O(n)$ utilizando $O(n)$-palavras de espaço.
- Construímos a tabela de índices em tempo $O(n)$ utilizando $O(n)$-bit de espaço.

Desafio:

1. É possível construir a tabela E obter um certificado em $O(n)$-bit de espaço?

Publicações

Blocos haplótipos e Indexação de strings

1. L. Cunha, Y. Diekmann, L. Kowada, Y. Stoye.
   Lecture Notes in Computer Science
   Brazilian Symposium on Bioinformatics (BSB'18).

2. L. Cunha, S. Dantas, T. Gagie, R. Wittler, L. Kowada, Y. Stoye.
   International Proceedings in Informatics
   Annual Symposium on Combinatorial Pattern Matching (CPM'17).

- University of Bielefeld, Alemanha:
    - prof. Jens Stoye
    - prof. Roland Wittler

Colaborações
Blocos haplótipos e Indexação de strings

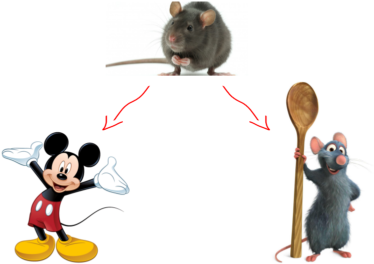
- University of Bielefeld, Alemanha:
  - prof. Jens Stoye
  - prof. Roland Wittler
- University College London, Inglaterra:
  - prof. Yoan Diekmann

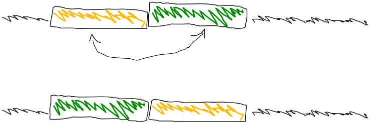
Colaborações

Blocos haplótipos e Indexação de strings

- University of Bielefeld, Alemanha:
  - prof. Jens Stoye
  - prof. Roland Wittler
- University College London, Inglaterra:
  - prof. Yoan Diekmann
- University Diego Portales, Chile:
  - prof. Travis Gagie

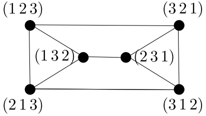
- University of Bielefeld, Alemanha:
  - prof. Jens Stoye
  - prof. Roland Wittler
- University College London, Inglaterra:
  - prof. Yoan Diekmann
- University Diego Portales, Chile:
  - prof. Travis Gagie
- Universidade Federal Fluminense:
  - profa. Simone Dantas
  - prof. Luis Kowada

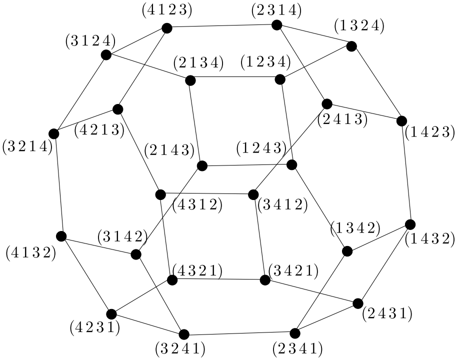




Ordenação por transposições (biológicas): Menor número de operações de transposições para transformar na identidade. Problema NP-completo.

Grafos de Cayley associados





- Melhores algoritmos:

- Melhores algoritmos:

Razão de aproximação

- Elias e Hartman, 2006: 1,375-aproximativo
  - complexidade: $O(n^2)$

- Melhores algoritmos:

Razão de aproximação

- Elias e Hartman, 2006: 1,375-aproximativo
  - complexidade: $O(n^2)$

Complexidade

- Feng e Zhu, 2007: Árvore de permutação
- Hartman e Shamir, 2005: 1,5-aproximativo
  - complexidade: $O(n \log n)$

- Melhores algoritmos:

Razão de aproximação

- Elias e Hartman, 2006: 1,375-aproximativo
  - complexidade: $O(n^2)$

Complexidade

- Feng e Zhu, 2007: Árvore de permutação
- Hartman e Shamir, 2005: 1,5-aproximativo
  - complexidade: $O(n \log n)$

- Por que não usar **Árvore de permutação** no **1,375-aproximativo**?
  - Proposta de Firoz *et al.* (*Journal of Comput. Biol.*, 2011).

- Melhores algoritmos:

Razão de aproximação

- Elias e Hartman, 2006: 1,375-aproximativo
  - complexidade: $O(n^2)$

Complexidade

- Feng e Zhu, 2007: Árvore de permutação
- Hartman e Shamir, 2005: 1,5-aproximativo
  - complexidade: $O(n \log n)$

- Por que não usar **Árvore de permutação** no **1,375-aproximativo**?
  - **Mostramos contra exemplos para a corretude.**

Novo algoritmo $1,375$-aproximativo

- Propomos um novo algoritmo $1,375$-aproximativo para obter complexidade $O(n \log n)$.
  - Adaptamos o algoritmo de Elias e Hartman e utilizamos a Árvore de permutação.

Novo algoritmo 1,375-aproximativo

- Propomos um novo algoritmo 1,375-aproximativo para obter complexidade $O(n \log n)$.
  - Adaptamos o algoritmo de Elias e Hartman e utilizamos a Árvore de permutação.

- Sempre encontramos uma $11/8$-sequência.
  - 11 operações tais que pelo menos 8 são boas.

Diâmetro

$D_M(n)$ = máxima distância pela métrica **M** para permutações de tamanho n.

Diâmetro

$D_M(n)$ = máxima distância pela métrica **M** para permutações de tamanho n.

Diâmetro de Transposição: $D_t(n) \geq \lfloor \frac{n+1}{2} \rfloor + 1$

- Proposta de limite $D_t(n) \geq \frac{17n+1}{33}$
  Lu e Yang, *SIAM J. Disc. Math.* 2010.

Diâmetro

$D_M(n)$ = máxima distância pela métrica **M** para permutações de tamanho n.

Diâmetro de Transposição: $D_t(n) \geq \lfloor \frac{n+1}{2} \rfloor + 1$

- Proposta de limite $D_t(n) \geq \frac{17n+1}{33}$
  Lu e Yang, *SIAM J. Disc. Math.* 2010.

1. Invalidamos esta proposta;
2. Mostramos novas famílias infinitas com distâncias iguais ao limite corrente;
3. Mostramos estratégia para obter $D_t(n) \geq \lfloor \frac{n+1}{2} \rfloor + 2$.
   Cunha, Hausen, Kowada, de Figueiredo, *SIAM J. Disc. Math.* 2013.

1. L. Cunha, L. Kowada, R. Hausen, C. de Figueiredo.
   Journal of Computational Biology, 2015.
2. L. Cunha, L. Kowada, C. de Figueiredo
   Matemática Contemporânea, 2015.
3. L. Cunha, L. Kowada, C. de Figueiredo.
   Cologne-Twente Workshop on Graphs & Combinatorial Optimization (CTW'15).
4. L. Cunha, L. Kowada, R. Hausen, C. de Figueiredo.
   Lecture Notes in Computer Science
   Workshop Algorithms on Bioinformatics (WABI'14).
5. L. Cunha, L. Kowada, R. Hausen, C. de Figueiredo.
   International Colloquium on Graph Theory and Combinatorics (ICGT'14).
6. L. Cunha, L. Kowada, R. Hausen, C. de Figueiredo.
   Lecture Notes in Computer Science
   Brazilian Symposium on Bioinformatics (BSB'13).
7. L. Cunha, L. Kowada, R. Hausen, C. de Figueiredo.
   SIAM Journal on Discrete Mathematics, 2013
8. L. Cunha, L. Kowada, R. Hausen, C. de Figueiredo.
   Lecture Notes in Computer Science,
   Brazilian Symposium on Bioinformatics (BSB'12).
9. L. Cunha, L. Kowada
   Matemática Contemporânea, 2010.

- Universidade Federal Fluminense:
  - prof. Luis Kowada

Colaborações

- Universidade Federal Fluminense:
  - prof. Luis Kowada
- Universidade Federal do Rio de Janeiro:
  - profa. Celina de Figueiredo

Colaborações

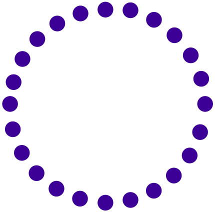
Ordenação de permutações

- Universidade Federal Fluminense:
  - prof. Luis Kowada
- Universidade Federal do Rio de Janeiro:
  - profa. Celina de Figueiredo
- Universidade Federal do ABC:
  - prof. Rodrigo Hausen

Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

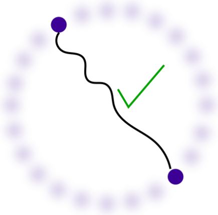
Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

Centralidade:



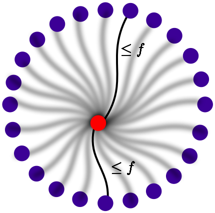
Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

Centralidade:



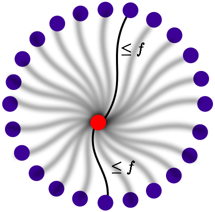
Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

Centralidade:



Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

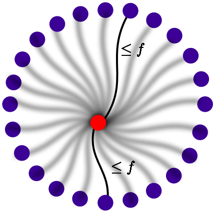
Centralidade:



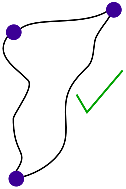
Mediana:

Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

Centralidade:

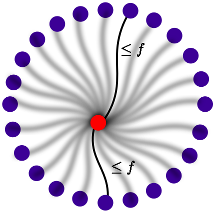


Mediana:

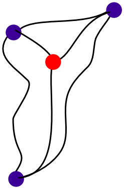


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

Centralidade:

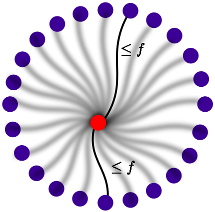


Mediana:

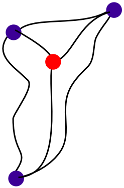


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

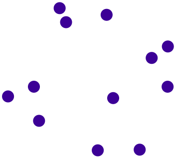
Centralidade:



Mediana:

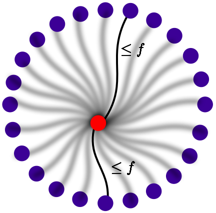


Convexidade:

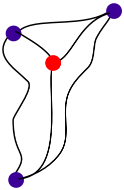


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

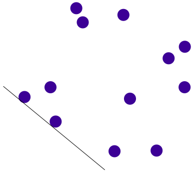
Centralidade:



Mediana:

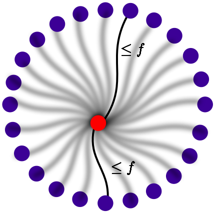


Convexidade:

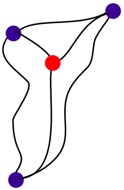


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

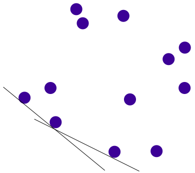
Centralidade:



Mediana:

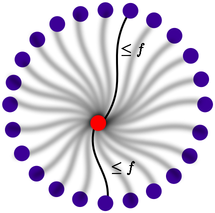


Convexidade:

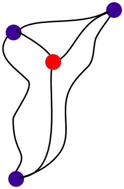


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

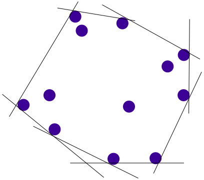
Centralidade:



Mediana:

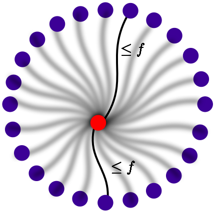


Convexidade:

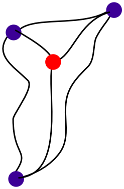


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

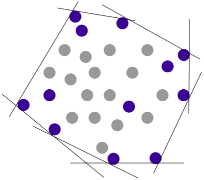
Centralidade:



Mediana:

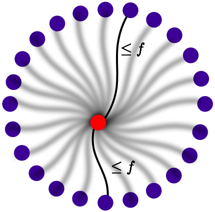


Convexidade:

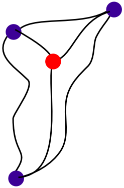


Apesar de Ordenação por transposições ser NP-completo, existem operações cujo problema de ordenação é polinomial. Porém, outros desafios surgem:

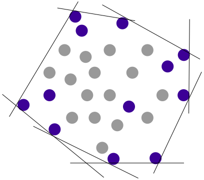
Centralidade:



Mediana:



Convexidade:



- **Resultados:** Propriedades, NP-completudes e algoritmos eficientes para muitos problemas nestes temas.

1. L. Cunha, F. Protti.
   Journal of Computational Biology, 2019
2. L. Cunha, P. Feijão, V. Santos, L. Kowada, C. de Figueiredo
   Discrete Applied Mathematics, 2019.
3. L. Cunha, F. Protti
   Electronic Notes in Discrete Matematics.
   International Conference on Applied Combinatorial Optimization (EURO/ALIO'18).
4. L. Cunha, V. Santos, L. Kowada, C. de Figueiredo
   Matemática Contemporânea, 2017.
5. L. Cunha, V. Santos, L. Kowada, C. de Figueiredo
   Congreso Latino-Iberoamericano de Investigación Operativa (CLAIO'16).
6. L. Cunha, V. Santos, L. Kowada, C. de Figueiredo
   Latin American Workshop on Cliques in Graphs (LAWCG'16).

- Simon Fraser University, Canadá:
  - prof. Pedro Feijão

- Simon Fraser University, Canadá:
  - prof. Pedro Feijão
- Universidade Federal Fluminense:
  - prof. Luis Kowada
  - prof. Fábio Protti

- Simon Fraser University, Canadá:
  - prof. Pedro Feijão
- Universidade Federal Fluminense:
  - prof. Luis Kowada
  - prof. Fábio Protti
- Universidade Federal do Rio de Janeiro:
  - profa. Celina de Figueiredo

- Simon Fraser University, Canadá:
  - prof. Pedro Feijão
- Universidade Federal Fluminense:
  - prof. Luis Kowada
  - prof. Fábio Protti
- Universidade Federal do Rio de Janeiro:
  - profa. Celina de Figueiredo
- Universidade Federal de Minas Gerais:
  - prof. Vinícius Santos

Atividades de ensino e orientações

Atividades de ensino

Há 8 anos mediador presencial do CEDERJ/UFF - polo Rio Bonito

Atividades de ensino e orientações

Atividades de ensino

Há 8 anos mediador presencial do CEDERJ/UFF - polo Rio Bonito

- Matemática:

  Análise Real

  Matemática Discreta

Atividades de ensino e orientações

Atividades de ensino

Há 8 anos mediador presencial do CEDERJ/UFF - polo Rio Bonito

- Matemática:
  - Análise Real
  - Matemática Discreta
- Computação:
  - Fundamentos de Algoritmos para Computação
  - Probabilidade e Estatística
  - Estruturas de Dados

Atividades de ensino e orientações

Atividades de ensino

Há 8 anos mediador presencial do CEDERJ/UFF - polo Rio Bonito

- Matemática:
  - Análise Real
  - Matemática Discreta
- Computação:
  - Fundamentos de Algoritmos para Computação
  - Probabilidade e Estatística
  - Estruturas de Dados

Orientações e contribuições

- Orientação de 3 TCC's (TSC/UFF)

Atividades de ensino e orientações

Atividades de ensino

Há 8 anos mediador presencial do CEDERJ/UFF - polo Rio Bonito

- Matemática:
  - Análise Real
  - Matemática Discreta
- Computação:
  - Fundamentos de Algoritmos para Computação
  - Probabilidade e Estatística
  - Estruturas de Dados

Orientações e contribuições

- Orientação de 3 TCC's (TSC/UFF)
- Colaboração em 1 doutorado

Atividades de ensino e orientações

Atividades de ensino

Há 8 anos mediador presencial do CEDERJ/UFF - polo Rio Bonito

- Matemática:
  - Análise Real
  - Matemática Discreta
- Computação:
  - Fundamentos de Algoritmos para Computação
  - Probabilidade e Estatística
  - Estruturas de Dados

Orientações e contribuições

- Orientação de 3 TCC's (TSC/UFF)
- Colaboração em 1 doutorado
- Colaboração em 2 IC's e 2 TCC's: profa. Simone Dantas (GAN/IME-UFF) e profa. Fernanda Couto (DCC/IM-UFRRJ)

Atividades de Extensão

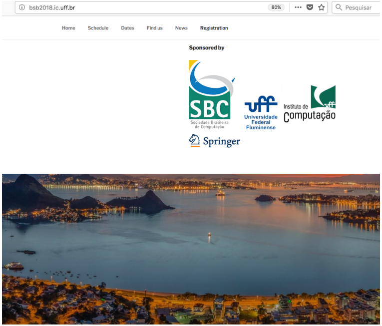
VIII Latin American Workshop on Cliques in Graphs

- Comitê de organização

Atividades de Extensão

XI Bazilian Symposium on Bioinformatics

- Comitê de programa e comitê de organização



Workshop Matemática Discreta e Aplicações

- Oficina: Propagação de Epidemia

- Desafios: Quão eficaz são as atividades lúdicas em escolas? Melhora concentração e pensamento abstrato?

Muito Obrigado!