Parameterized Complexity of Critical Node Cuts

Celso Aimbiré Weffort Santos

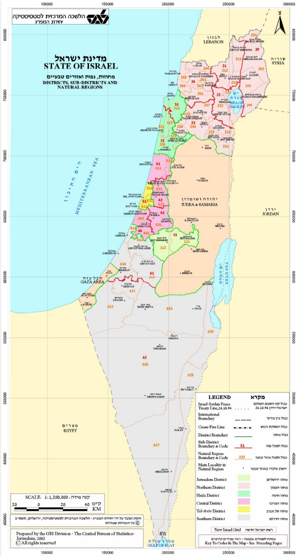
Universidade Estadual de Campinas

celso.santos@students.ic.unicamp.br

15 de dezembro de 2016

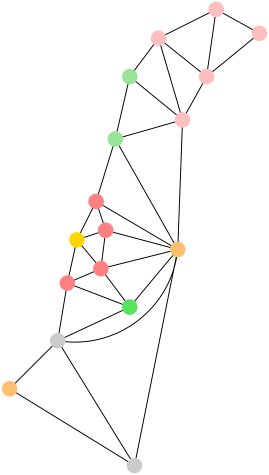
O Problema

O Problema

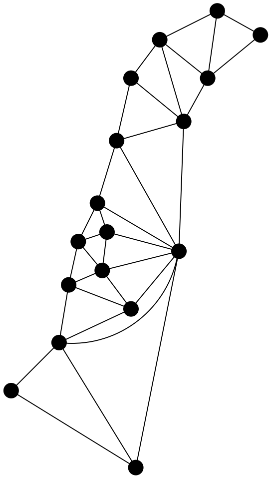


מדינת ישראל
STATE OF ISRAEL
DISTRICTS, SUB-DISTRICTS AND NATURAL REGIONS
LEBANON
SYRIA
JORDAN
EGYPT
GAZA AREA
JUDEA & SAMARIA
MEDITERRANEAN SEA
DEAD SEA
GULF OF ELAT
LEGEND
מקרא
SCALE: 1:1,200,000
KM
Prepared by the GIS Division - The Central Bureau of Statistics-Jerusalem, 2002
©All rights reserved

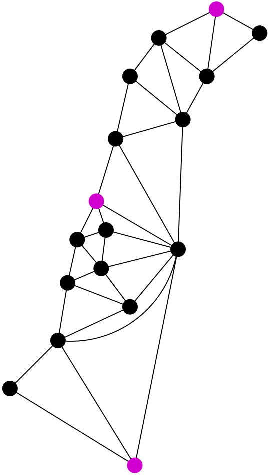
O Problema



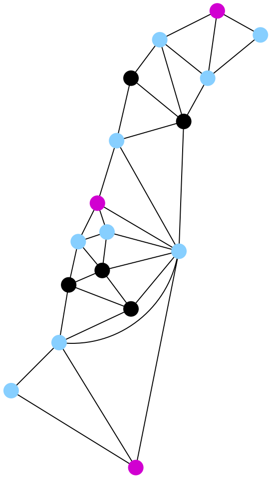
O Problema



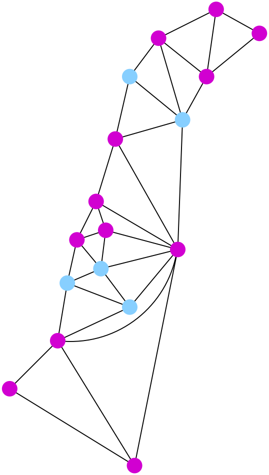
O Problema

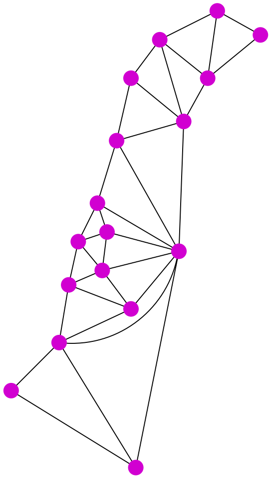


O Problema



O Problema





Critical Node Cut (CNC)

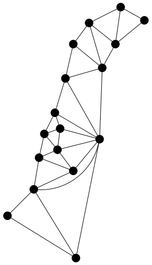
Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

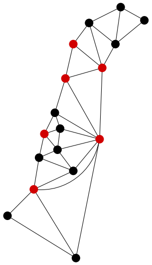
Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?



Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?



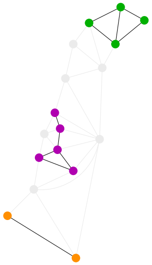
Parâmetros:
$k = 6$
$x = 34$

Critical Node Cut

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

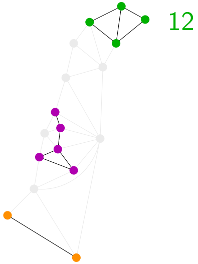


Parâmetros:
$k = 6$
$x = 34$

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

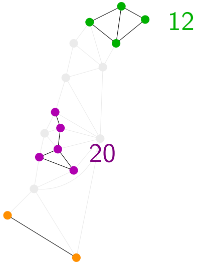


Parâmetros:
$k = 6$
$x = 34$

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

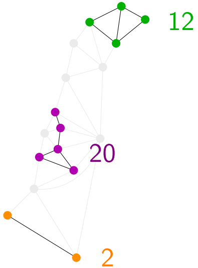


Parâmetros:
$k = 6$
$x = 34$

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

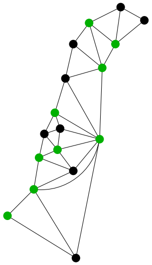


Parâmetros:
$k = 6$
$x = 34$

CRITICAL NODE CUT (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

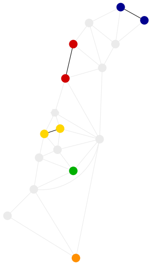


Parâmetros:
$k = 9$
$x = 6$

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

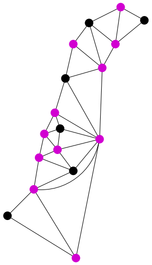


Parâmetros:
$k = 9$
$x = 6$

CRITICAL NODE CUT (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?

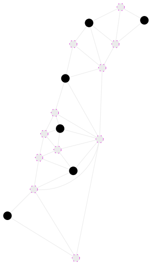


Parâmetros:
$k = 11$
$x = 0$

Critical Node Cut (CNC)

Entrada: Um grafo $G = (V, E)$ e $k, x \in \mathbb{N}$

Problema: Existe $C \subseteq V(G)$, com $|C| \leq k$, tal que $G - C$ possui no máximo x pares conectados de vértices?



Parâmetros:
$k = 11$
$x = 0$

Possíveis Parâmetros

– O tamanho k do corte C

Possíveis Parâmetros

- O tamanho k do corte C
- O número máximo x de pares conectados de vértices em $G - C$

Possíveis Parâmetros

- O tamanho k do corte C
- O número máximo x de pares conectados de vértices em $G - C$
- O número y de pares conectados de vértices que devem ser removidos de G

Possíveis Parâmetros

- O tamanho k do corte C
- O número máximo x de pares conectados de vértices em $G - C$
- O número y de pares conectados de vértices que devem ser removidos de G
- O *treewidth* w de G

Possíveis Parâmetros

- O tamanho k do corte C
- O número máximo x de pares conectados de vértices em $G - C$
- O número y de pares conectados de vértices que devem ser removidos de G
- O *treewidth* w de G
- Combinações entre estes parâmetros

Resultados do Artigo

Parâmetro				Resultado	
k	x	y	w	FPT	P-*kernel*
✓				*não*	*não*
	✓			*não*	*não*
		✓		*sim*	*não*
			✓	*não*	*não*
✓	✓			*sim*	*sim*
✓		✓		*sim*	*não*
✓			✓	**?**	*não*
	✓	✓		*sim*	*sim*
	✓		✓	*sim*	*não*
		✓	✓	*sim*	*não*
✓	✓	✓		*sim*	*sim*
✓	✓		✓	*sim*	*sim*
✓		✓	✓	*sim*	*não*
	✓	✓	✓	*sim*	*sim*
✓	✓	✓	✓	*sim*	*sim*

Tabela: Resultados do artigo *Parameterized compelexity of critical node cuts*

Resultados Analisados

Parâmetro				**Resultado**	
k	x	y	w	FPT	P-*kernel*
✓				*não*	*não*
		✓		*sim*	*não*
✓	✓			*sim*	*sim*

Parâmetro $k + x$

Seja (G, k, x) uma instância de CNC

Seja (G, k, x) uma instância de CNC

Reduções:

Seja (G, k, x) uma instância de CNC

Reduções:

CNC.1 Se existe vértice $v \in V(G)$, tal que $d(v) = 0$, remova v. A nova instância é $(G - v, k, x)$.

Seja (G, k, x) uma instância de CNC

Reduções:

CNC.1 Se existe vértice $v \in V(G)$, tal que $d(v) = 0$, remova v. A nova instância é $(G - v, k, x)$.

CNC.2 Se existe vértice $v \in V(G)$, tal que $d(v) > k + \sqrt{x}$, remova v. A nova instância é $(G - v, k - 1, x)$.

Seja (G, k, x) uma instância de CNC

Reduções:

CNC.1 Se existe vértice $v \in V(G)$, tal que $d(v) = 0$, remova v. A nova instância é $(G - v, k, x)$.

CNC.2 Se existe vértice $v \in V(G)$, tal que $d(v) > k + \sqrt{x}$, remova v. A nova instância é $(G - v, k - 1, x)$. (Exercício)

Seja (G, k, x) uma instância de CNC

Reduções:

CNC.1 Se existe vértice $v \in V(G)$, tal que $d(v) = 0$, remova v. A nova instância é $(G - v, k, x)$.

CNC.2 Se existe vértice $v \in V(G)$, tal que $d(v) > k + \sqrt{x}$, remova v. A nova instância é $(G - v, k - 1, x)$.

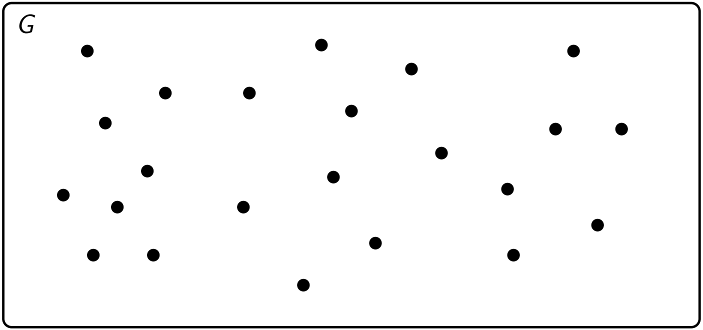
Lema 1

Se as reduções CNC.1 e CNC.2 não são aplicáveis, então

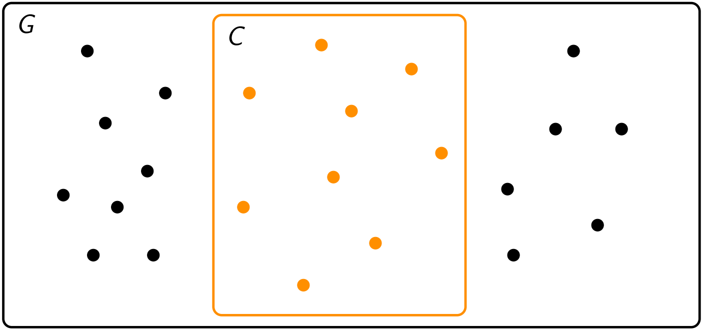
$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$

Prova

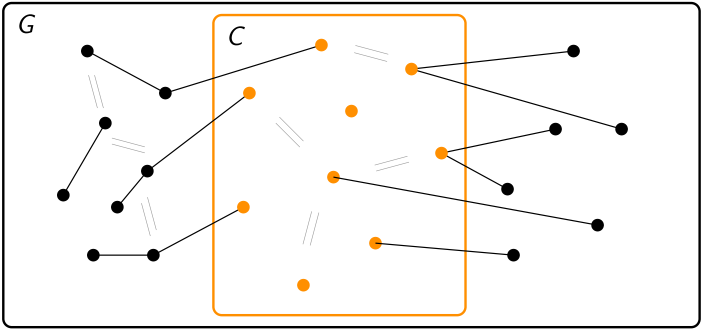
$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$



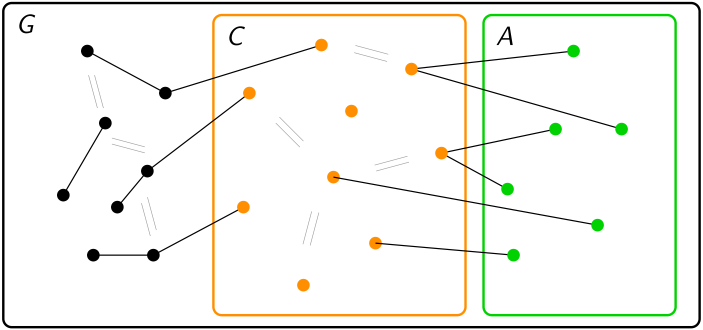
$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$



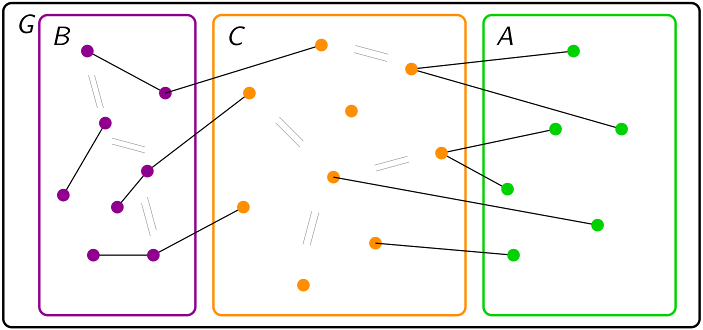
$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$



$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$

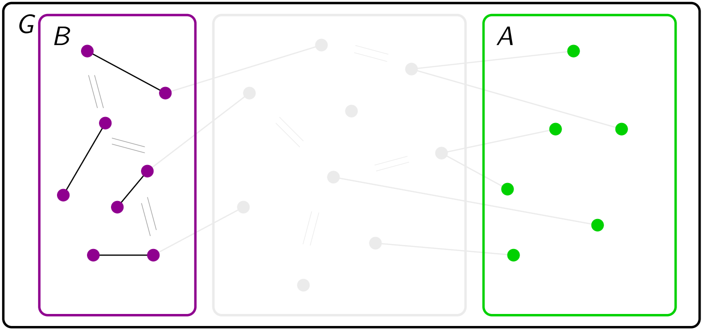


$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$

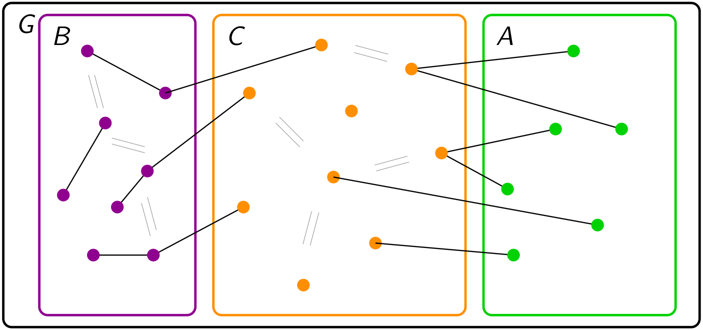


Prova

$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$



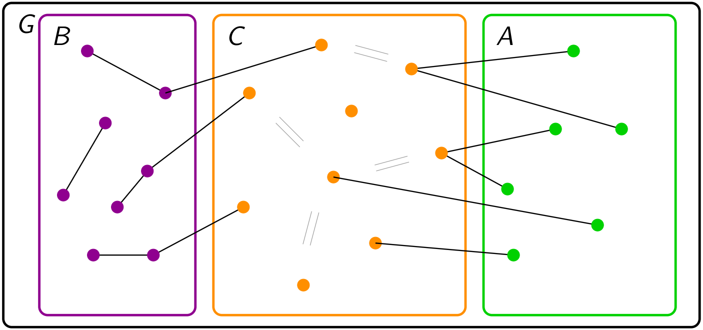
$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$



$$|C| \leq k$$

Prova

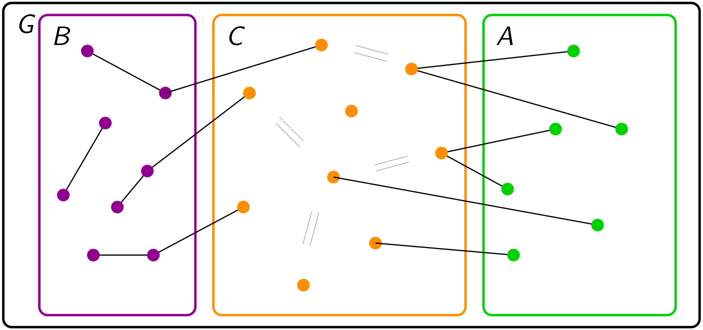
$$|V(G)| \leq k(k + \sqrt{x}) + x + k$$



$$|B| \leq x$$

Prova

$$|V(G)| \leq k(k+\sqrt{x}) + x + k$$



$$|A| \leq k(k+\sqrt{x})$$

Seja (G, k, x) uma instância de CNC

Seja (G, k, x) uma instância de CNC

Suponha que exista corte C de G de tamanho k, tal que $G - C$ possui, no máximo, x pares conectados de vértices

Seja (G, k, x) uma instância de CNC

Suponha que exista corte C de G de tamanho k, tal que $G - C$ possui, no máximo, x pares conectados de vértices

$$|E(G - C)| \leq x$$

Seja (G, k, x) uma instância de CNC

Suponha que exista corte C de G de tamanho k, tal que $G - C$ possui, no máximo, x pares conectados de vértices

$$|E(G - C)| \leq x$$

Problema Auxiliar:
Determinar se existem k vértices em $C' \subseteq V(G)$, tal que $G - C'$ tem, no máximo, x arestas

Algoritmo FPT

Seja (G, k, x) uma instância de CNC

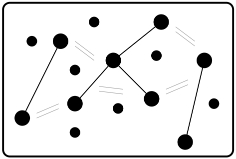
Suponha que exista corte C de G de tamanho k, tal que $G - C$ possui, no máximo, x pares conectados de vértices

$$|E(G - C)| \leq x$$

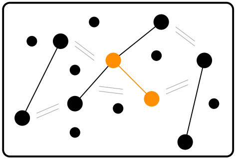
Problema Auxiliar:
Determinar se existem k vértices em $C' \subseteq V(G)$, tal que $G - C'$ tem, no máximo, x arestas

Bounded Search Trees

(G, k, x)

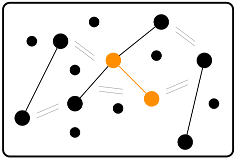


(G, k, x)

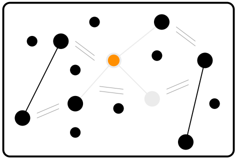


e = uv

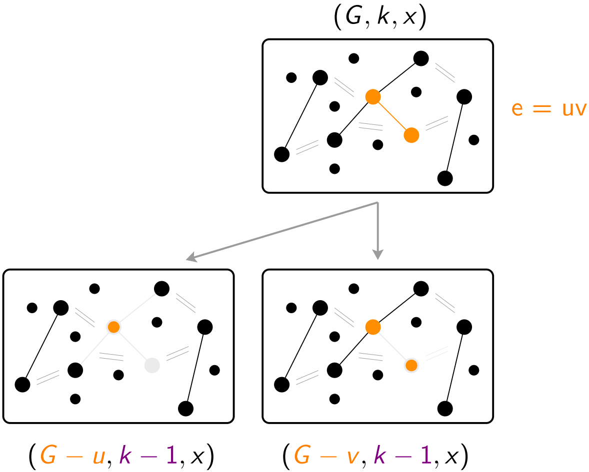
(G, k, x)



$e = uv$

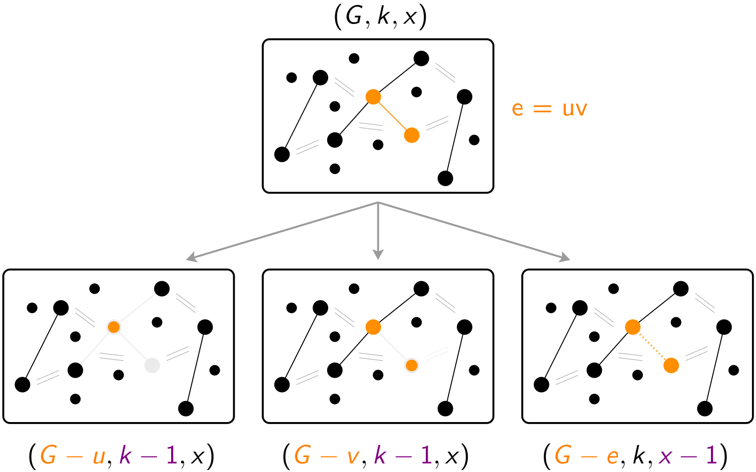


$(G - u, k - 1, x)$



(G, k, x)
$e = uv$
$(G - u, k - 1, x)$
$(G - v, k - 1, x)$

Algoritmo FPT



Algoritmo FPT

- Enumeramos todas as possíveis soluções mínimas do problema auxiliar $\rightarrow$ $\mathcal{C}'$

Algoritmo FPT

- Enumeramos todas as possíveis soluções mínimas do problema auxiliar $\rightarrow$ $\mathcal{C}'$
- Se existe solução C para nossa instância de CNC, então C também é uma solução para o PA

Algoritmo FPT

- Enumeramos todas as possíveis soluções mínimas do problema auxiliar $\rightarrow$ $\mathcal{C}'$
- Se existe solução C para nossa instância de CNC, então C também é uma solução para o PA
- Para cada $C' \in \mathcal{C}'$, verificamos se é possível gerar uma solução de CNC

Algoritmo FPT

- Enumeramos todas as possíveis soluções mínimas do problema auxiliar $\rightarrow$ $\mathcal{C}'$
- Se existe solução C para nossa instância de CNC, então C também é uma solução para o PA
- Para cada $C' \in \mathcal{C}'$, verificamos se é possível gerar uma solução de CNC
- Se não for possível para todo C', então estamos tratando de uma instância *não*

Problema auxiliar: $3^{x+k} \cdot n$

Tempo de Execução

Problema auxiliar: $3^{x+k} \cdot n$

Processar solução mínima:

Problema auxiliar: $3^{x+k} \cdot n$

Processar solução mínima:

Suponha que $C' \in \mathcal{C}'$, com $|C'| = k_1$

Problema auxiliar: $3^{x+k} \cdot n$

Processar solução mínima:

Suponha que $C' \in \mathcal{C}'$, com $|C'| = k_1$

$k_2 = k - k_1$

Problema auxiliar: $3^{x+k} \cdot n$

Processar solução mínima:

Suponha que $C' \in \mathcal{C}'$, com $|C'| = k_1$

$k_2 = k - k_1$

Existem, no máximo, $2x$ vértices

Problema auxiliar: $3^{x+k} \cdot n$

Processar solução mínima:

Suponha que $C' \in \mathcal{C}'$, com $|C'| = k_1$

$k_2 = k - k_1$

Existem, no máximo, $2x$ vértices

$\mathcal{O}(\binom{2x}{k_2}x^2 + n)$

Problema auxiliar: $3^{x+k} \cdot n$

Processar solução mínima:

Suponha que $C' \in \mathcal{C}'$, com $|C'| = k_1$

$k_2 = k - k_1$

Existem, no máximo, $2x$ vértices

$\mathcal{O}(\binom{2x}{k_2}x^2 + n)$

Tempo total: $\mathcal{O}(3^{x+k}(x^{k+2} + n))$

Parâmetro y

Parâmetro y

y é o número de pares conectados de vértices a serem removidos de G

Parâmetro y

y é o número de pares conectados de vértices a serem removidos de G

$$y = n(n-1) - k$$

y é o número de pares conectados de vértices a serem removidos de G

$$y = n(n-1) - k$$

Teorema 2

Critical Node Cut é FPT quando parametrizado por y.

Parâmetro y

y é o número de pares conectados de vértices a serem removidos de G

$$y = n(n-1) - k$$

Teorema 2

CRITICAL NODE CUT é FPT quando parametrizado por y.

Programação Dinâmica

Parâmetro y

Seja (G, k, y) uma instância de CNC

Parâmetro y

Seja (G, k, y) uma instância de CNC
$G_1, \ldots, G_t$ as componentes conexas de G

Parâmetro y

Seja (G, k, y) uma instância de CNC
$G_1, \ldots, G_t$ as componentes conexas de G

$|V(G_i)| \geq y$ ✓

Seja (G, k, y) uma instância de CNC
$G_1, \ldots, G_t$ as componentes conexas de G

$|V(G_i)| \geq y$ ✓

$k \geq y$ ✓

Seja (G, k, y) uma instância de CNC
$G_1, \ldots, G_t$ as componentes conexas de G

$|V(G_i)| \geq y \quad \checkmark$

$k \geq y \quad \checkmark$

$k < y$

1. Para cada G_i e cada $1 \leq k' \leq k$:
   O número máximo de pares conectados de vértices destruídos ao remover k' vértices $\quad \rightarrow \quad T[i, k']$

Algoritmo FPT

1. Para cada G_i e cada $1 \leq k' \leq k$:
   O número máximo de pares conectados de vértices destruídos ao remover k' vértices $\quad \rightarrow \quad T[i, k']$

2. Para i crescente:
   o número de pares conectados destruídos ao remover k' vértices de $G_1, \ldots, G_i \quad \rightarrow \quad Q[i, k']$

1. Para cada G_i e cada $1 \leq k' \leq k$:
   O número máximo de pares conectados de vértices destruídos ao remover k' vértices $\rightarrow$ $T[i, k']$

2. Para i crescente:
   o número de pares conectados destruídos ao remover k' vértices de $G_1, \ldots, G_i$ $\rightarrow$ $Q[i, k']$

   $i = 1 \rightarrow Q[1, k'] = T[1, k']$

Algoritmo FPT

1. Para cada G_i e cada $1 \leq k' \leq k$:
   O número máximo de pares conectados de vértices destruídos ao remover k' vértices $\quad \rightarrow \quad T[i, k']$

2. Para i crescente:
   o número de pares conectados destruídos ao remover k' vértices de $G_1, \ldots, G_i \quad \rightarrow \quad Q[i, k']$

$$i = 1 \rightarrow Q[1, k'] = T[1, k']$$
$$i > 1 \rightarrow Q[i, k'] = \max_{k'' \leq k'} Q[i-1, k''] + T[i, k' - k'']$$

Algoritmo FPT

1. Para cada G_i e cada $1 \leq k' \leq k$:
   O número máximo de pares conectados de vértices destruídos ao remover k' vértices $\quad \rightarrow \quad T[i, k']$

2. Para i crescente:
   o número de pares conectados destruídos ao remover k' vértices de $G_1, \ldots, G_i \quad \rightarrow \quad Q[i, k']$

$$i = 1 \rightarrow Q[1, k'] = T[1, k']$$
$$i > 1 \rightarrow Q[i, k'] = \max_{k'' \leq k'} Q[i-1, k''] + T[i, k' - k'']$$

3. Se $Q[t, k] < y$, devolvemos *não*; caso contrário, devolvemos *sim*

Tempo de Execução

Cada componente conexa: 2^y

Tempo de Execução

Cada componente conexa: 2^y

Calcular o tamanho de cada componente conexa restante: $\mathcal{O}(y^2)$

Tempo de Execução

Cada componente conexa: 2^y

Calcular o tamanho de cada componente conexa restante: $\mathcal{O}(y^2)$

Programação dinâmica: $k^2 \leq y^2$ combinações de k' e k''

Tempo de Execução

Cada componente conexa: 2^y

Calcular o tamanho de cada componente conexa restante: $\mathcal{O}(y^2)$

Programação dinâmica: $k^2 \leq y^2$ combinações de k' e k''

Tempo total: $\mathcal{O}(2^y \cdot y^2 \cdot n)$

Parâmetro k

Teorema 3

Critical Node Cut é W[1]-difícil quando parametrizado por k.

Teorema 3

Critical Node Cut é W[1]-difícil quando parametrizado por k.

Reduzimos Clique para CNC.

Teorema 3

CRITICAL NODE CUT é W[1]-difícil quando parametrizado por k.

Reduzimos CLIQUE para CNC.

CLIQUE

Entrada:	Um grafo $G = (V, E)$ e $k \in \mathbb{N}$
Problema:	Existe um subgrafo de G isomorfo ao grafo K_k?

Teorema 3

CRITICAL NODE CUT é W[1]-difícil quando parametrizado por k.

Reduzimos CLIQUE para CNC.

CLIQUE

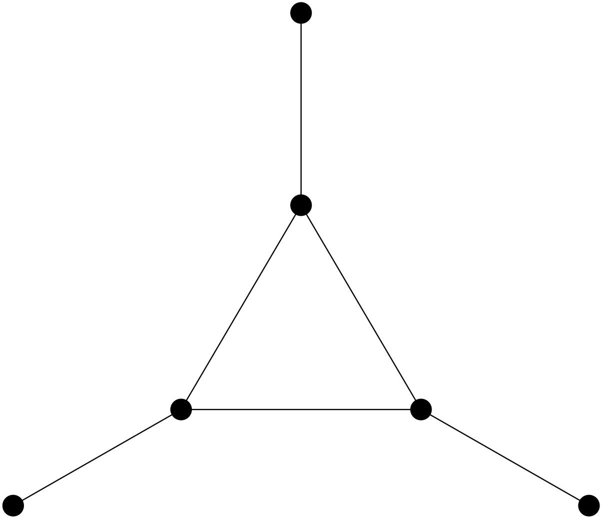
Entrada:	Um grafo $G = (V, E)$ e $k \in \mathbb{N}$
Problema:	Existe um subgrafo de G isomorfo ao grafo K_k?

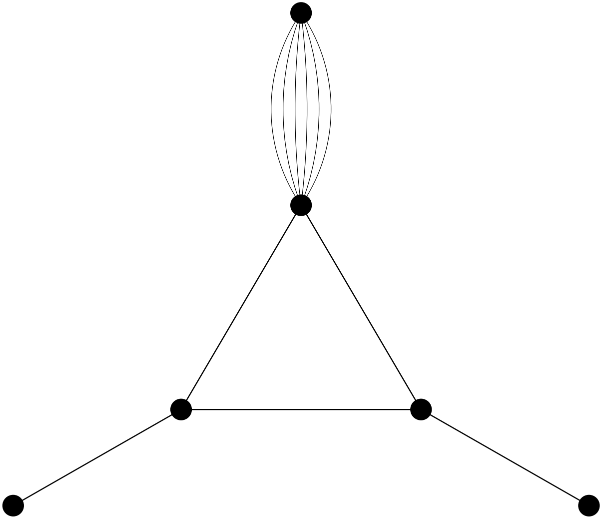
Seja (G, k) uma instância de CLIQUE. Construímos uma instância (H, l) de CNC, tal que G possui uma clique de l vértices se e somente se o grafo H tem $k = l$ vértices cuja remoção destrói

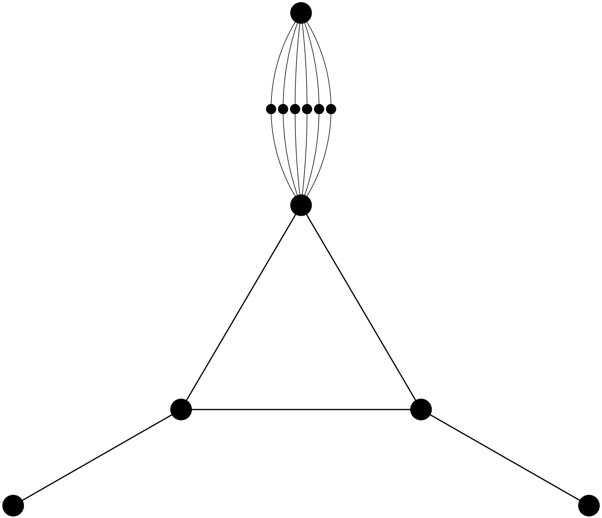
$$y = k(k-1) + 2k(N-k) + \binom{k}{2}n \cdot (\binom{k}{2}n - 1) + 2\binom{k}{2}n \cdot (N - k - \binom{k}{2}n)$$

pares conectados de vértices de H, onde $N := |V(H)|$

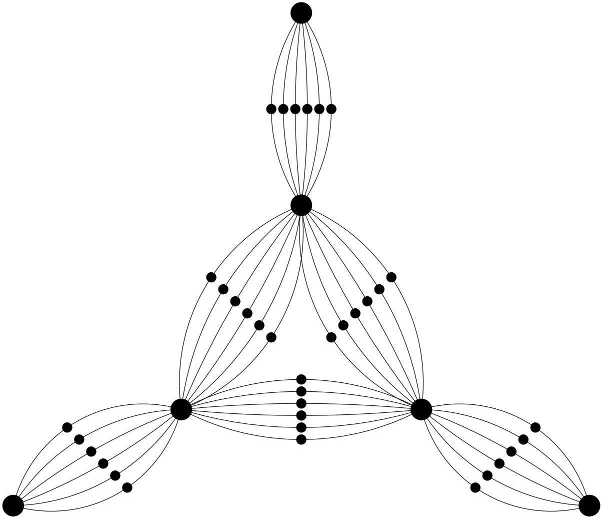
Construção de H



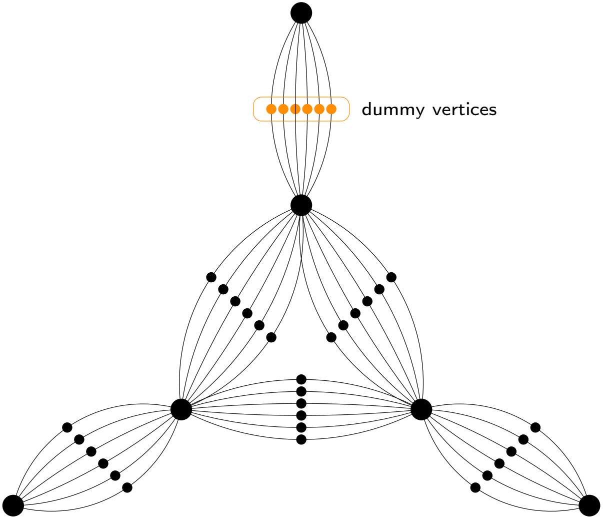




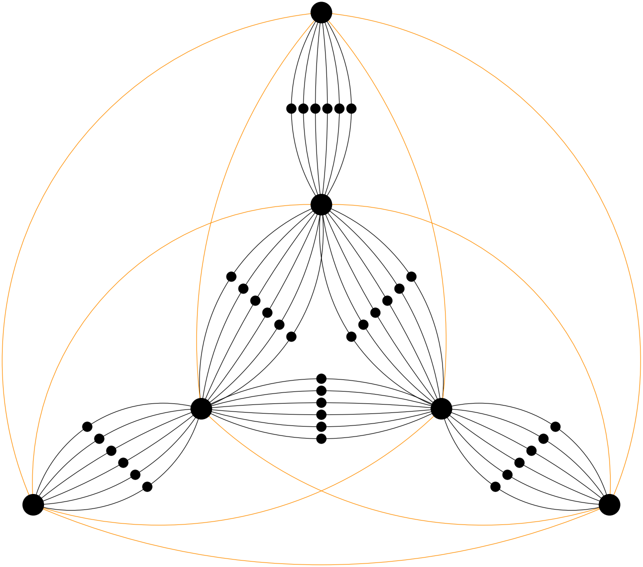
Construção de H



Construção de H

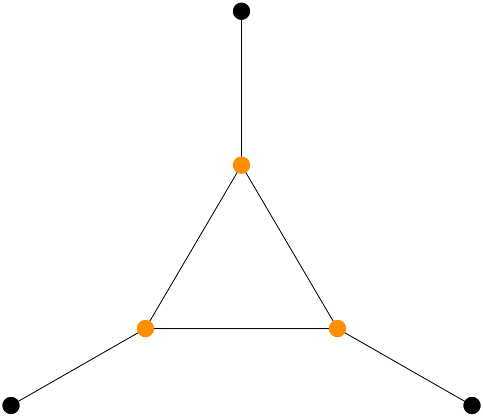


Construção de H



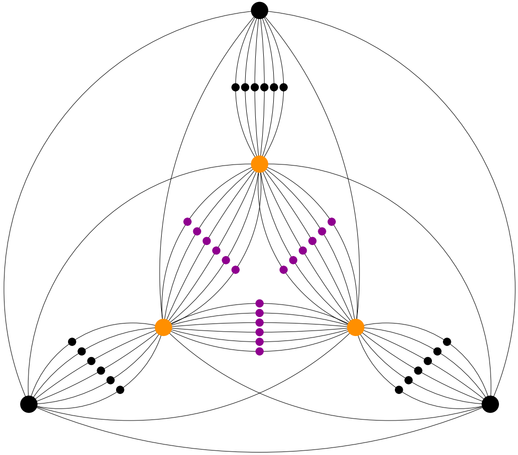
Prova ($\Rightarrow$)

Suponha que exista uma clique C em G de tamanho l.



Suponha que exista uma clique C em G de tamanho l.

Denotamos por $D(C)$ os $\binom{k}{2}n$ *dummy vertices* com ambos os vizinhos em C.

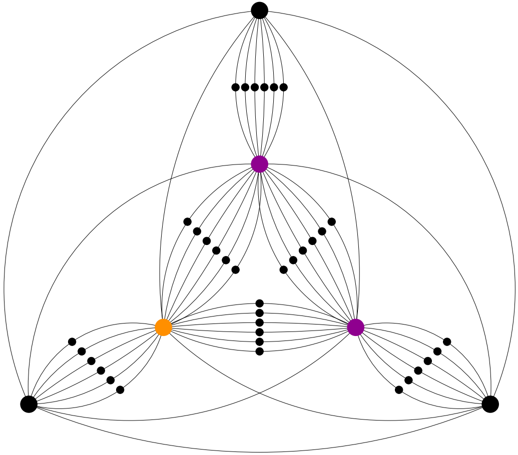


Suponha que exista uma clique C em G de tamanho l.

Denotamos por $D(C)$ os $\binom{k}{2} n$ *dummy vertices* com ambos os vizinhos em C.

Ao remover C de H, destruímos:

- $k(k-1)$



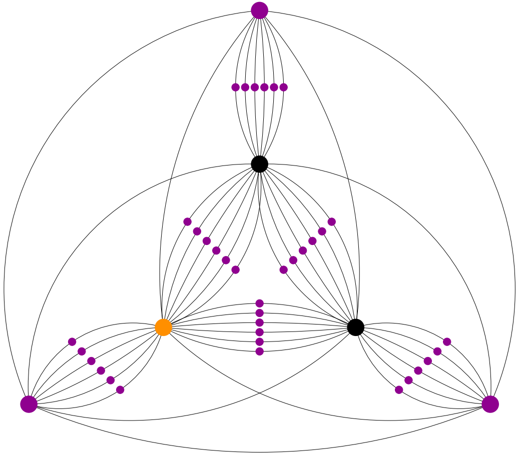
Prova ($\Rightarrow$)

Suponha que exista uma clique C em G de tamanho l.

Denotamos por $D(C)$ os $\binom{k}{2}n$ *dummy vertices* com ambos os vizinhos em C.

Ao remover C de H, destruímos:

- $k(k-1)$
- $2k(N-k)$



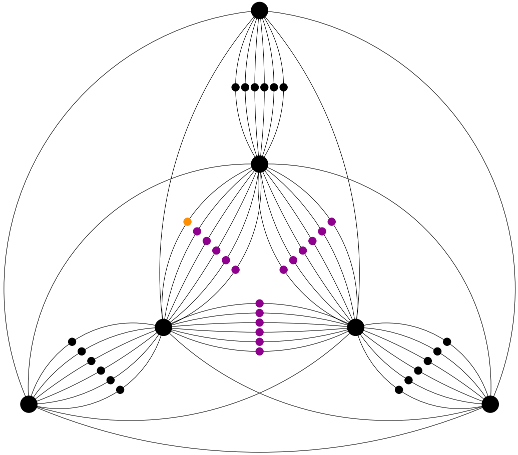
Prova ($\Rightarrow$)

Suponha que exista uma clique C em G de tamanho l.

Denotamos por $D(C)$ os $\binom{k}{2}n$ *dummy vertices* com ambos os vizinhos em C.

Ao remover C de H, destruímos:

- $k(k-1)$
- $2k(N-k)$
- $\binom{k}{2}n \cdot (\binom{k}{2}n - 1)$

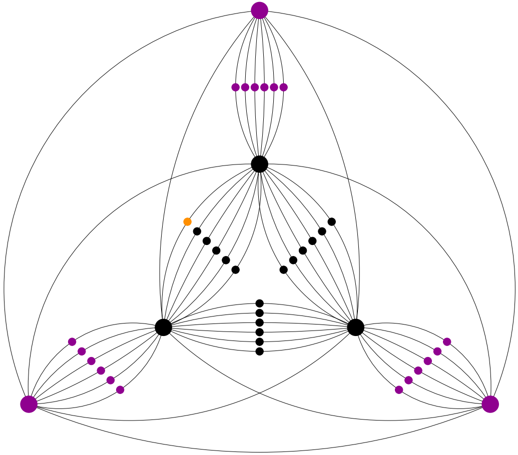


Suponha que exista uma clique C em G de tamanho l.

Denotamos por $D(C)$ os $\binom{k}{2}n$ *dummy vertices* com ambos os vizinhos em C.

Ao remover C de H, destruímos:

- $k(k-1)$
- $2k(N-k)$
- $\binom{k}{2}n \cdot (\binom{k}{2}n - 1)$
- $2\binom{k}{2}n \cdot (N - k - \binom{k}{2}n)$



Prova ($\Leftarrow$)

Suponha que exista um corte C que remove y pares conectados de vértices em H.

Prova ($\Leftarrow$)

Suponha que exista um corte C que remove y pares conectados de vértices em H.

Podemos supor que C é composto inteiramente por *non-dummy vertices*.

Prova ($\Leftarrow$)

Suponha que exista um corte C que remove y pares conectados de vértices em H.

Podemos supor que C é composto inteiramente por *non-dummy vertices*.

Ao remover um vértice v de C, removemos apenas pares conectados que envolvem v ou os *dummy vertices* vizinhos de v.

Prova ($\Leftarrow$)

Suponha que exista um corte C que remove y pares conectados de vértices em H.

Podemos supor que C é composto inteiramente por *non-dummy vertices*.

Ao remover um vértice v de C, removemos apenas pares conectados que envolvem v ou os *dummy vertices* vizinhos de v.

Removendo um corte C de tamanho k, removemos exatamente $k(k-1) + 2k(N-k)$ pares conectados.

Prova ($\Leftarrow$)

Suponha que exista um corte C que remove y pares conectados de vértices em H.

Podemos supor que C é composto inteiramente por *non-dummy vertices*.

Ao remover um vértice v de C, removemos apenas pares conectados que envolvem v ou os *dummy vertices* vizinhos de v.

Removendo um corte C de tamanho k, removemos exatamente $k(k-1) + 2k(N-k)$ pares conectados.

Para destruir todos os y pares conectados, precisamos isolar exatamente $\binom{k}{2}n$ *dummy vertices*.

Prova ($\Leftarrow$)

Suponha que exista um corte C que remove y pares conectados de vértices em H.

Podemos supor que C é composto inteiramente por *non-dummy vertices*.

Ao remover um vértice v de C, removemos apenas pares conectados que envolvem v ou os *dummy vertices* vizinhos de v.

Removendo um corte C de tamanho k, removemos exatamente $k(k-1) + 2k(N-k)$ pares conectados.

Para destruir todos os y pares conectados, precisamos isolar exatamente $\binom{k}{2}n$ *dummy vertices*.

Isso é feito encontrando k vértices que são adjacentes dois-a-dois por *edge-gadgets*. $\square$

Exercício

Exercício

Prove que a *Redução CNC.2* é segura.

Redução CNC.2 Se existe vértice $v \in V(G)$, tal que $d(v) > k + \sqrt{x}$, remova v. A nova instância é $(G - v, k - 1, x)$.