Algoritmos Parametrizados

Miscelânea

Lehilton Pedrosa

Segundo Semestre de 2016

Instituto de Computação – Unicamp

Roteiro

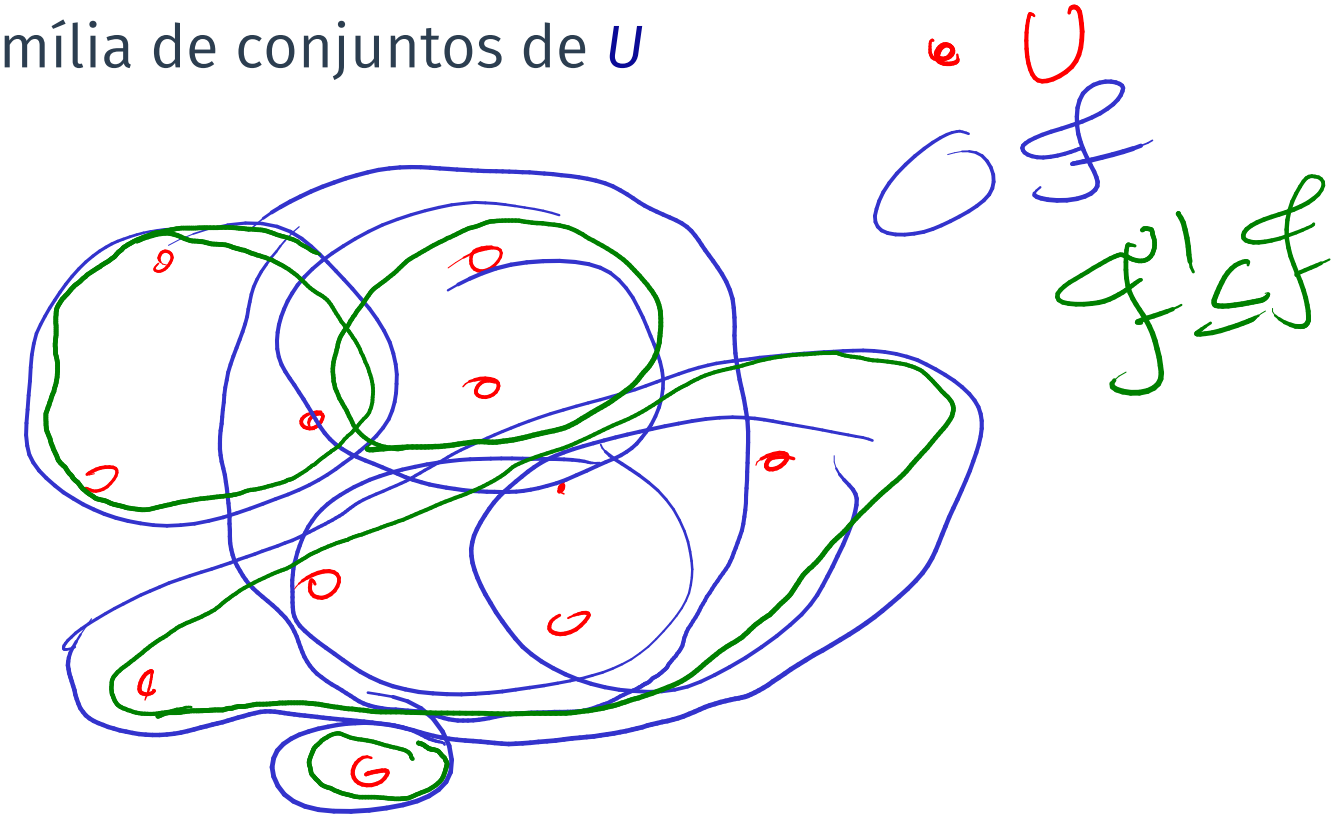
1. Programação Dinâmica sobre Subconjuntos
2. PLI de viabilidade
3. Teorema de Robertson e Seymour

Programação Dinâmica sobre Subconjuntos

Problema da cobertura de Conjuntos

Elementos:

- seja U um conjunto finito
- e $\mathcal{F}$ uma família de conjuntos de U



Problema da cobertura de Conjuntos

Elementos:

- seja U um conjunto finito
- e $\mathcal{F}$ uma família de conjuntos de U

Dado $U' \subseteq U$ e $\mathcal{F}' \subseteq \mathcal{F}$, dizemos que $\mathcal{F}'$ cobre U' se

$$\bigcup_{F \in \mathcal{F}'} F = U'.$$

Problema da cobertura de Conjuntos

Elementos:

parâmetro $k = |U|$

- seja U um conjunto finito
- e $\mathcal{F}$ uma família de conjuntos de U

Dado $U' \subseteq U$ e $\mathcal{F}' \subseteq \mathcal{F}$, dizemos que $\mathcal{F}'$ cobre U' se

$$\bigcup_{F \in \mathcal{F}'} F = U'.$$

Problema do Cobertura de Conjuntos Mínima

Dado universo U e família de conjuntos $\mathcal{F}$, queremos encontrar $\mathcal{F}' \subseteq \mathcal{F}$ de tamanho mínimo que cobre U.

Subproblema

Subproblema

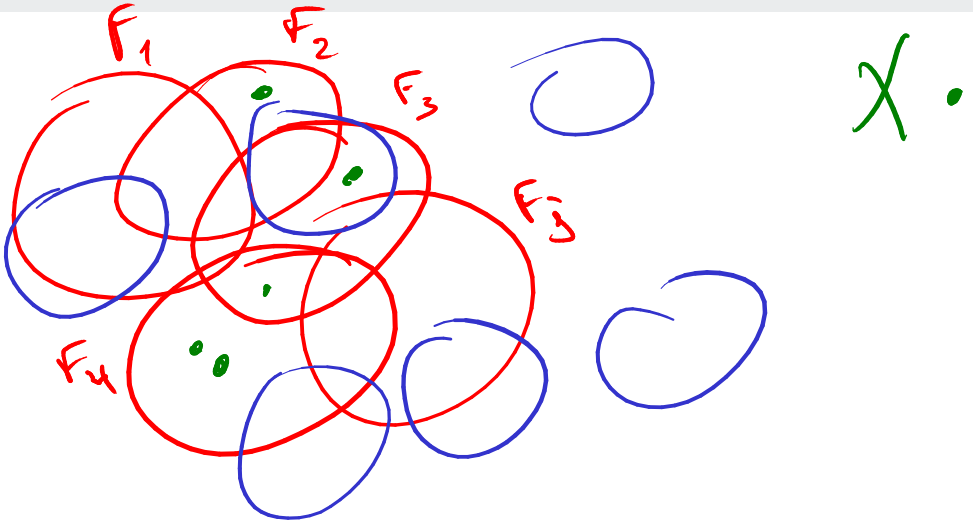
Seja $\mathcal{F} = \{F_1, F_2, \ldots, F_{|\mathcal{F}|}\}$ uma ordenação de $\mathcal{F}$ e seja $\mathcal{F}_j = \{F_1, F_2, \ldots, F_j\}$, para $0 \leq j \leq |\mathcal{F}|$.

Subproblema

Subproblema

Seja $\mathcal{F} = \{F_1, F_2, \ldots, F_{|\mathcal{F}|}$ uma ordenação de $\mathcal{F}$ e seja $\mathcal{F}_j = \{F_1, F_2, \ldots, F_j\}$, para $0 \leq j \leq |\mathcal{F}|$.

Dados $X \subseteq U$ e $0 \leq j \leq |\mathcal{F}|$, encontre um conjunto $\mathcal{F}' \subseteq F_j$ de tamanho mínimo que cobre X.



$j = 5$

$\mathcal{F}_5 = \{F_1, F_2, \ldots, F_5\}$

$\mathcal{F}' = \{F_2, F_3, F_4\}$

Subproblema

Subproblema

Seja $\mathcal{F} = \{F_1, F_2, \ldots, F_{|\mathcal{F}|}\}$ uma ordenação de $\mathcal{F}$ e seja $\mathcal{F}_j = \{F_1, F_2, \ldots, F_j\}$, para $0 \leq j \leq |\mathcal{F}|$.

Dados $X \subseteq U$ e $0 \leq j \leq |\mathcal{F}|$, encontre um conjunto $\mathcal{F}' \subseteq F_j$ de tamanho mínimo que cobre X.

Defina

$$T[X, j] = \begin{cases} |\mathcal{F}'| & \text{se houver tal } \mathcal{F}', \\ \infty & \text{c.c.} \end{cases}$$

Programação dinâmica

Teorema

Existe algoritmo que, dados U e $\mathcal{F} \subseteq U$:

- *encontra uma cobertura de U de tamanho mínimo;*
- *executa em tempo $2^{|U|}(|\mathcal{F}| + |U|)^{\mathcal{O}(1)}$*

Prova Algoritmo de Prog. Din:

a) Caso base: Se $j = 0$

Sabemos q $\$_0 = \phi$, então:

$$T[X, 0] = \begin{cases} 0 & , \text{se } X = \phi \\ \infty & , \text{c.c} \end{cases}$$

b) caso geral: Se $j > 0$

Calculamos:

$$T[X, j] = \min\{T[X, j-1], T[X \setminus F_j, j-1] + 1\}$$

Af $T[X, j]$ está bem definido

i) Mostrar que $T[X, j] \geq \min\{T[X \setminus F_j, j-1] + 1, T[X, j-1]\}$

· Seja $\mathcal{F}'$ uma solução de (X, j)

1 - Se $F_j \in \mathcal{F}' \Rightarrow \bigcup_{S \in \mathcal{F}' \setminus \{F_j\}} S \cup F_j \supseteq X$

$$\Rightarrow \bigcup_{S \in \mathcal{F}' \setminus \{F_j\}} S \supseteq X \setminus F_j$$

Daí, $\mathcal{F}' \setminus \{F_j\}$ é solução p/ $(X \setminus F_j, j-1)$.

$$\Rightarrow T[X \setminus F_j, j-1] \leq |\mathcal{F}'| - 1 = T[X, j] - 1$$

2 - Se $F_j \notin \mathcal{F}' \Rightarrow \bigcup_{S \in \mathcal{F}' \setminus \{F_j\}} S = X$

Como $\mathcal{F} \setminus \{F_j\} \subseteq \mathcal{F}_{j-1} \Rightarrow T[X, j-1] \leq |\mathcal{F}'|$

$= T[X, j-1]$

ii) Mostrar: $T[X,j] \leq \min\{T[X,j-1], T[X \setminus F_j, j-1] + 1\}$

1 - Se $T[X,j-1]$ é menor:

Seja $\mathcal{F}''$ uma solução de $(X, j-1)$

Mas $\mathcal{F}'' \subseteq \mathcal{F}_{j-1} \subseteq \mathcal{F}_j$

$\Rightarrow \mathcal{F}''$ é viável para (X, j)

$\Rightarrow T[X,j] \leq |\mathcal{F}''| = T[X,j-1]$.

2 - Se $1 + T[X \setminus F_j, j-1]$ é menor:

Seja $\mathcal{F}''$ uma solução de $(X \setminus F_j, j-1)$

Temos $\bigcup_{S \in \mathcal{F}''} S \supseteq X \setminus F_j$

$$\Rightarrow \left(\bigcup_{S \in \mathcal{F}''} S\right) \cup F_j \supseteq (X \setminus F_j) \cup F_j \supseteq X$$

$\Rightarrow T[X,j] \leq |\mathcal{F}''| + 1 = T[X \setminus F_j, j-1] + 1$.

Quanto tempo gasta a PD?

$$\#\text{conf.} = 2^{|U|} \cdot (|S| + 1)$$

$$\text{Tempo/conf} = (|U| + |S|)^{O(1)}$$

$$\times$$

$$2^{|U|} \cdot (|U| + |S|)^{O(1)}$$

Problema da Árvore de Steiner

Seja G um grafo e $K \subseteq V(G)$. Uma árvore de Steiner é um subgrafo H conexo acíclico de G tal que $K \subseteq V(H)$.

Problema da Árvore de Steiner

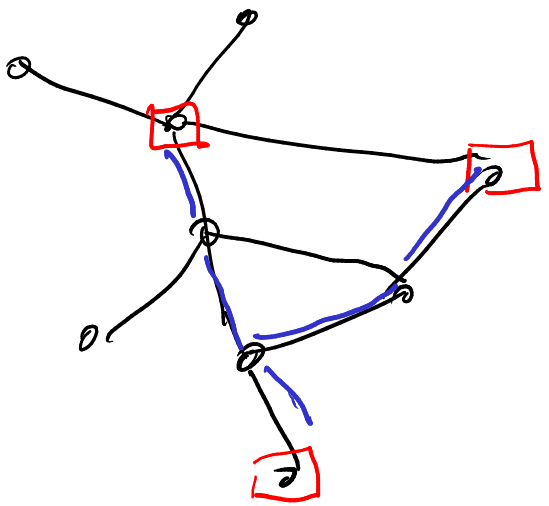
Seja G um grafo e $K \subseteq V(G)$. Uma árvore de Steiner é um subgrafo H conexo acíclico de G tal que $K \subseteq V(H)$.

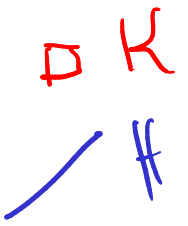
- K são chamados de terminais

Problema da Árvore de Steiner

Seja G um grafo e $K \subseteq V(G)$. Uma árvore de Steiner é um subgrafo H conexo acíclico de G tal que $K \subseteq V(H)$.

- K são chamados de terminais
- $V(G) \setminus K$ são chamados de vértices de Steiner





Problema da Árvore de Steiner

Seja G um grafo e $K \subseteq V(G)$. Uma árvore de Steiner é um subgrafo H conexo acíclico de G tal que $K \subseteq V(H)$.

- K são chamados de terminais
- $V(G) \setminus K$ são chamados de vértices de Steiner

Problema da Árvore de Steiner Mínima

Dados grafo G, conjunto de terminais $K \subseteq V(G)$ e custos $w : E(G) \to \mathbb{R}_{>0}$, queremos encontrar uma árvore de Steiner H cujo custo $w(E(H))$ é mínimo.

Problema da Árvore de Steiner

Seja G um grafo e $K \subseteq V(G)$. Uma árvore de Steiner é um subgrafo H conexo acíclico de G tal que $K \subseteq V(H)$.

- K são chamados de terminais
- $V(G) \setminus K$ são chamados de vértices de Steiner

Problema da Árvore de Steiner Mínima

Dados grafo G, conjunto de terminais $K \subseteq V(G)$ e custos $w : E(G) \to \mathbb{R}_{>0}$, queremos encontrar uma árvore de Steiner H cujo custo $w(E(H))$ é mínimo.

- se $w(e) = 1$ para $e \in E(G)$, então o equivale a minimizar $|V(H)|$ ou $|E(H)|$.

Problema da Árvore de Steiner

Seja G um grafo e $K \subseteq V(G)$. Uma árvore de Steiner é um subgrafo H conexo acíclico de G tal que $K \subseteq V(H)$.

- K são chamados de terminais
- $V(G) \setminus K$ são chamados de vértices de Steiner

Problema da Árvore de Steiner Mínima

Dados grafo G, conjunto de terminais $K \subseteq V(G)$ e custos $w : E(G) \to \mathbb{R}_{>0}$, queremos encontrar uma árvore de Steiner H cujo custo $w(E(H))$ é mínimo.

- se $w(e) = 1$ para $e \in E(G)$, então o equivale a minimizar $|V(H)|$ ou $|E(H)|$.
- $\text{dist}(u, v)$ é o custo de um caminho mínimo de u até v.

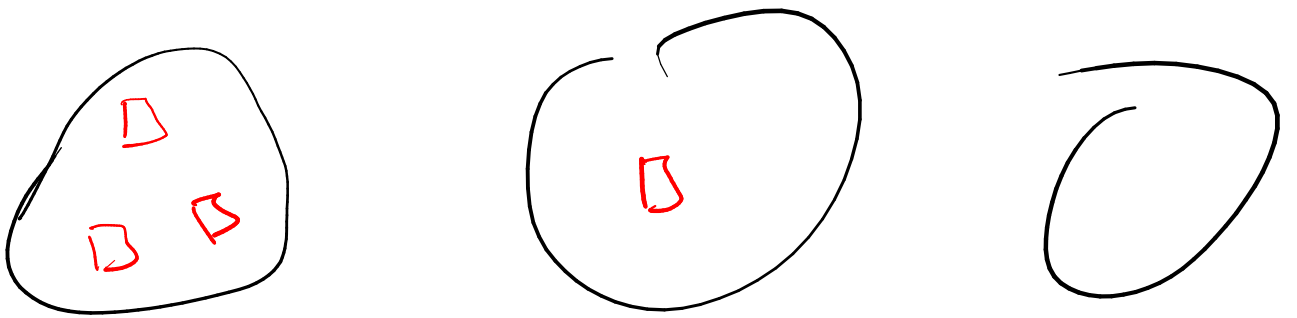
Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:



Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.

Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Redução 3: Se $t \in K$ com $d(t) \geq 1$,

Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Redução 3: Se $t \in K$ com $d(t) \geq 1$,então modifique a instância adicionando um novo vértice t' e aresta tt' e faça

Preprocessando

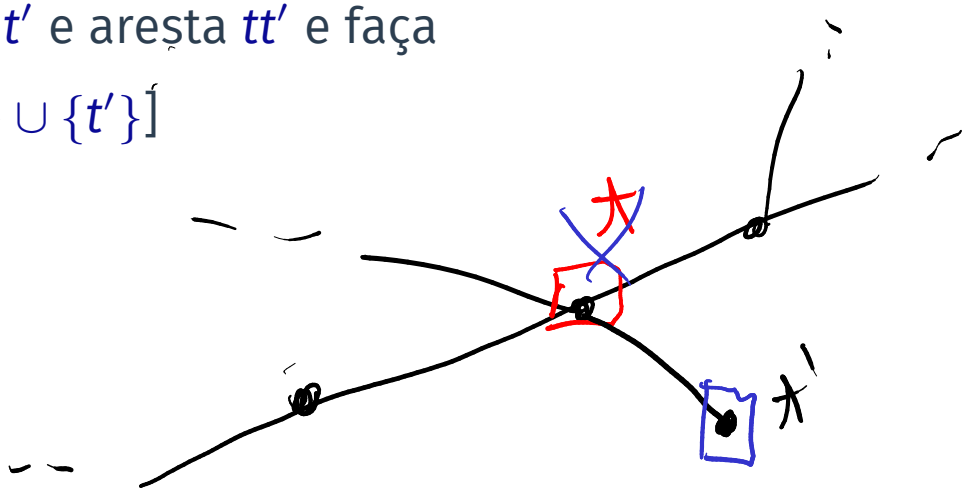
Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Redução 3: Se $t \in K$ com $d(t) \geq 1$,então modifique a instância adicionando um novo vértice t' e aresta tt' e faça

- $K \leftarrow K \setminus \{t\} \cup \{t'\}$
- $w(t, t') = 1$



Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Redução 3: Se $t \in K$ com $d(t) \geq 1$,então modifique a instância adicionando um novo vértice t' e aresta tt' e faça

- $K \leftarrow K \setminus \{t\} \cup \{t'\}]$
- $w(t, t') = 1$

Observações:

Preprocessando

Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Redução 3: Se $t \in K$ com $d(t) \geq 1$,então modifique a instância adicionando um novo vértice t' e aresta tt' e faça

- $K \leftarrow K \setminus \{t\} \cup \{t'\}]$
- $w(t, t') = 1$

Observações:

- cada aplicação aumenta 1 ao custo da árvore mínima

Preprocessando

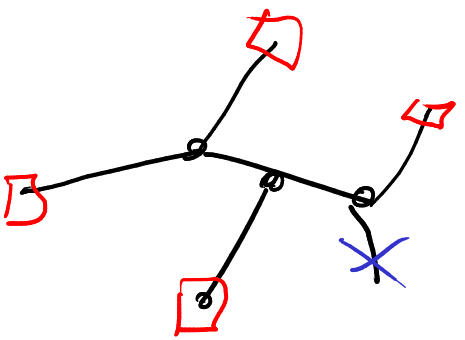
Redução 1: Se $|K| \leq 1$, então resolva de maneira ótima em tempo polinomial.

Redução 2: Se G é desconexo então:

- se existe componente conexa C de G tal que $V(C) \subseteq K$, devolva a instância C, K, w.
- caso contrário, conclua que não há solução.

Redução 3: Se $t \in K$ com $d(t) \geq 1$,então modifique a instância adicionando um novo vértice t' e aresta tt' e faça

- $K \leftarrow K \setminus \{t\} \cup \{t'\}]$
- $w(t, t') = 1$



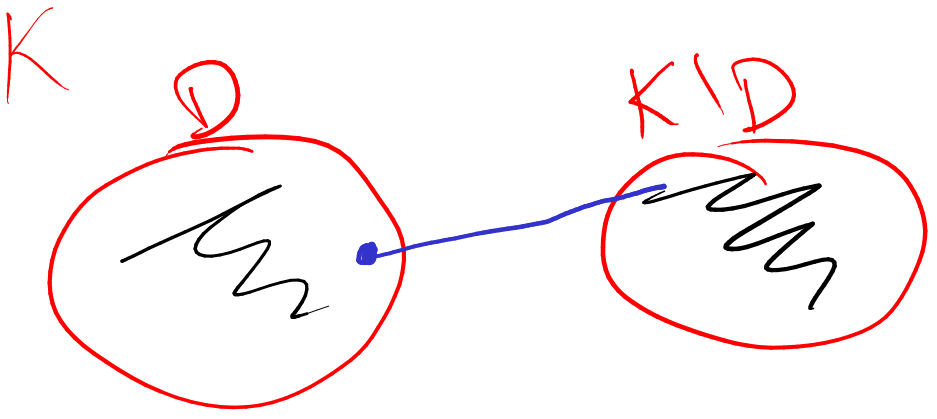
Observações:

- cada aplicação aumenta 1 ao custo da árvore mínima
- em uma árvore mínima, os terminais são folhas

Subproblema

Subproblema

Dados $D \subseteq K$ e $v \in V \setminus K$, encontre uma árvore de Steiner H mínima com terminais $D \cup \{v\}$.



Subproblema

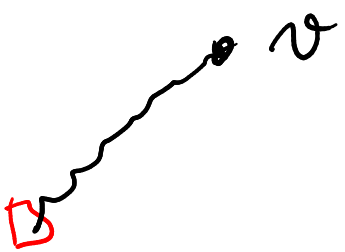
Subproblema

Dados $D \subseteq K$ e $v \in V \setminus K$, encontre uma árvore de Steiner H mínima com terminais $D \cup \{v\}$.

Defina

$$T[D, v] = w(H).$$

Caso base: $|D| = 1$

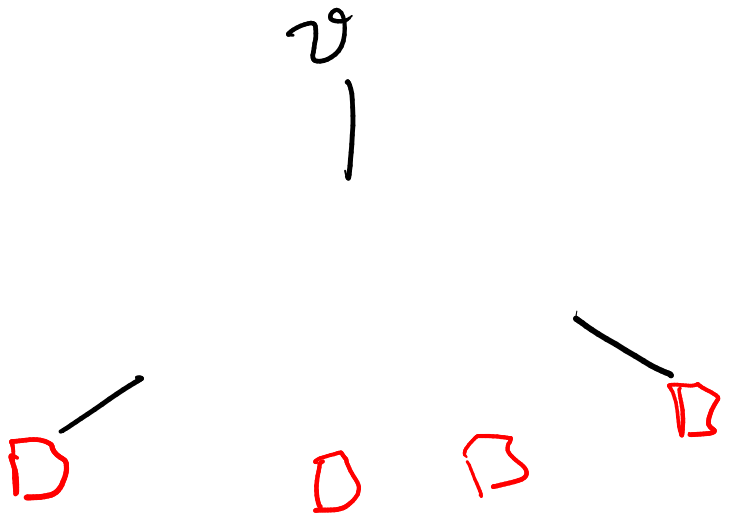


Se $D = \{t\}$, faça $T[D, v] = dist(t, v)$

Resolvendo subproblema

Lema

Seja $D \subseteq K$, tal que $|D| \geq 2$ e $v \in V(G) \setminus K$,

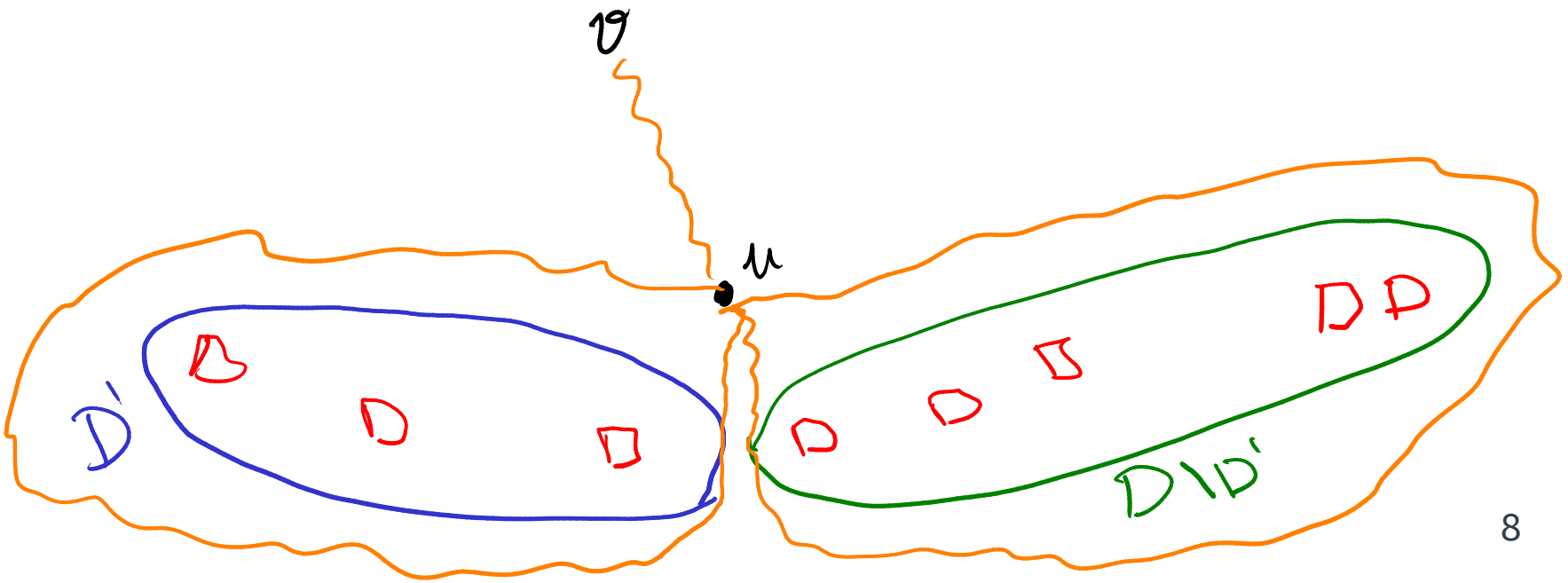


Resolvendo subproblema

Lema

Seja $D \subseteq K$, tal que $|D| \geq 2$ e $v \in V(G) \setminus K$, então

$$T[D, v] = \min_{\substack{u \in V(G) \setminus K \\ \emptyset \neq D' \subsetneq D}} \{T[D', u] + T[D \setminus D', u] + dist(u, v)\}$$



Prova

① $T[D,v] \leq \min_{D',u} \{ T[D',u] + T[D \setminus D', u] + dist(u,v) \}$

· Seja H_1 a sol. ótima p/ (D', u)

· Seja H_2 a sol. ótima p/ $(D \setminus D', u)$

· Seja P um caminho min de u a v.

Temos q $H' = H_1 \cup H_2 \cup P$ é um grafo conexo.

Seja T' uma árvore ger. min de H'.

Como $V[T'] \supseteq D \cup \{v\} \Rightarrow T'$ é sol p/ (D, v)

$\Rightarrow T[D',u] + T[D \setminus D', u] + dist(u,v) =$

$w(H_1) + w(H_2) + w(P) \geq$

$w(H') \geq w(T') \geq T[D,v]$.

(ii) $T[D,v] \geq \min_{D',u} \{ T[D',u] + T[D \setminus D', u] + dist(u,v) \}$

Seja H uma árvore de Steiner mínima com $D \cup \{v\}$

Enraize H em v.

Seja $u \in V(H)$ tq:

(1) u tem pelo menos 2 filhos.

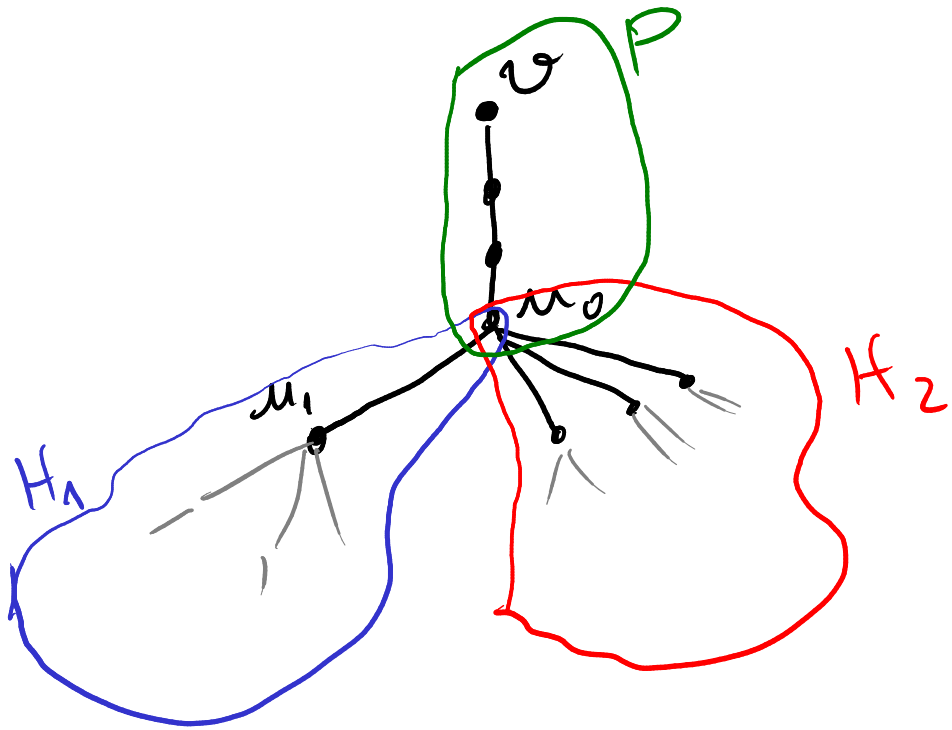
(2) seja o vértice mais próximo de v com (1).

O vértice u está bem definido pois H tem pelo menos 2 folhas em D, ou seja, descendentes de v.

Escolha u_1 um filho arbitrário de u_0.

Decomponha H em:

1) P: um caminho de v a u_0
2) H_1: a subárvore enraizada em u_1 mais (u_0, u_1)
3) H_2: a subárvore enraizada em u_0 menos H_1



- Seja $D' = V[H_1] \cap K$

- Temos que $D \setminus D' = V[H_2] \cap K$, pois $P \cap D = \emptyset$, já que terminais são folhas.

- Sabemos que $|D'| \geq 1$, pois H é ótima também $|D \setminus D'| \geq 1$.

- Note que $w(H_1) = T[D', u_0]$ pois H é ótima e $w(H_2) = T[D \setminus D', u_0]$.

Ainda $dist(v, u_0) = w(P)$.

Portanto:

$T[D, v] = w(H) = w(P) + w(H_1) + w(H_2) \geq$

$dist(u_0, v) + T[D', u_0] + T[D \setminus D', u_0]$ □.

Programação dinâmica

Teorema

Existe algoritmo que, dados G e $K \subseteq V(G)$:

- *encontra uma árvore de Steiner de custo mínimo;*
- *executa em tempo $3^{|K|} N^{\mathcal{O}(1)}$.*

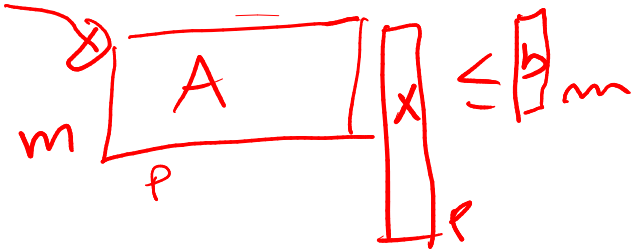
$$\text{Tempo} \in \sum_{v \in V \setminus K} \sum_{D \subseteq K} 2^{|D|} \cdot n^{\mathcal{O}(1)} \leq n \cdot n^{\mathcal{O}(1)} \cdot \sum_{j=1}^{|K|} \binom{|K|}{j} 2^j$$

$$= n^{\mathcal{O}(1)} \left(2+1\right)^{|K|}$$

$$\leq n^{\mathcal{O}(1)} \, 3^{|K|} \quad \square.$$

PLI de viabilidade

Relembrando



PLI de viabilidade

O problema de Programação Inteira de viabilidade é o problema de decidir se existe vetor $x \in \mathbb{Z}^p$ tal que

$$Ax \leq b,$$

onde A é uma matriz $m \times p$ e b um vetor com m linhas.

PLI é FPT

Tamanho de um PLI: o tamanho L de um PLI é o número de bits necessários para representá-lo.

PLI é FPT

Tamanho de um PLI: o tamanho L de um PLI é o número de bits necessários para representá-l.

O teorema abaixo é clássico de Lenstra Jr. e outros:

Teorema

PLI de viabilidade pode ser resolvido em tempo $\mathcal{O}(p^{2,5p+o(p)}L)$ com memória $L^{\mathcal{O}(1)}$, onde p é o número de variáveis e L é o número de bits do PLI.

⇒ queremos reescrever um problema como um PLI em que o nº de variáveis $p = f(k)$

Adicionando otimização

PLI

Em um PLI (de otimização), queremos encontrar o vetor $x \in \mathbb{Z}^p$ tal que $Ax \leq b$ e cujo valor de cx é mínimo, onde c é um vetor de p colunas.

Adicionando otimização

PLI

Em um PLI (de otimização), queremos encontrar o vetor $x \in \mathbb{Z}^p$ tal que $Ax \leq b$ e cujo valor de cx é mínimo, onde c é um vetor de p colunas.

- cx é chamada função objetivo

Resolvendo PLI

Resolvendo PLI

Teorema

PLI ser resolvido em tempo
$\mathcal{O}(p^{2,5p+o(p)}(L + \log M_x) \log(M_x M_c))$, *onde:*

Resolvendo PLI

$x_1 = 2^{100} \Rightarrow M_x \geq 2^{100}$

$10^{10} x_1 - 10^{100} x_2 \leq 3$

$\Rightarrow M_c = 10^{100}$

Teorema

PLI ser resolvido em tempo $\mathcal{O}(p^{2,5p+o(p)}(L + \log M_x)\log(M_x M_c))$, *onde:*

- p *é o número de variáveis*
- L *é o número de bits do PLI*
- M_x *é o maior valor absoluto de uma variável em solução ótima*
- M_c *é o maior valor absoluto de um coeficiente do PLI*

Prova

- Seja x^* uma sol ótima e $SOL = |cx^*|$.

$$SOL = |c_1 x_1 + c_2 x_2 + \dots + c_p x_p| \leq$$
$$|c_1 x_1| + |c_2 x_2| + \dots + |c_p x_p| \leq p \cdot \max_{i=1..p} |c_p \cdot x_p|$$
$$\leq p \cdot M_x \cdot M_c$$

- Busca com faz. buscabinário o valor da SOL.
- Sup. que cx^* é positivo.
  - adiciona $cx \leq t$ para dado t
  - Resolvemos o problema de viabilidade
    - a) Se for viável, diminuímos t
    - b) Se for inviável, aumentamos t

□

Problema do Desbalanceamento

Seja um grafo G e uma ordenação $\pi : V(G) \to [n]$:

Problema do Desbalanceamento

Seja um grafo G e uma ordenação $\pi : V(G) \to [n]$:

- Os predecessores e sucessores de v são:
  - $L_\pi(v) = \{u \in N(v) : \pi(u) < \pi(v)\}$,
  - $R_\pi(v) = \{u \in N(v) : \pi(u) > \pi(v)\}$.

Problema do Desbalanceamento

Seja um grafo G e uma ordenação $\pi : V(G) \to [n]$:

- Os predecessores e sucessores de v são:
  - $L_\pi(v) = \{u \in N(v) : \pi(u) < \pi(v)\}$,
  - $R_\pi(v) = \{u \in N(v) : \pi(u) > \pi(v)\}$.
- O desbalanço de um vértice v é $\iota_\pi(v) = \big||L_\pi(v)| - |R_\pi(v)|\big|$.

v

$L_\pi(v_4) = \{v_2, v_3\}$

$R_\pi(v_4) = \{v_5, v_7, v_6\}$

$\iota_\pi(v_4) = |2 - 3| = 1$

Problema do Desbalanceamento

Seja um grafo G e uma ordenação $\pi : V(G) \to [n]$:

- Os predecessores e sucessores de v são:
  - $L_\pi(v) = \{u \in N(v) : \pi(u) < \pi(v)\}$,
  - $R_\pi(v) = \{u \in N(v) : \pi(u) > \pi(v)\}$.
- O desbalanço de um vértice v é $\iota_\pi(v) = \big||L_\pi(v)| - |R_\pi(v)|\big|$.

Problema do Desbalanceamento

Dado um grafo G, encontre uma ordenação π que minimize

$$\sum_{v \in V(G)} \iota_\pi(v).$$

Parametrizando com cobertura por vértices

Ideia:

Parametrizando com cobertura por vértices

Ideia:

- Partimos de uma cobertura por vértices X de tamanho k

Parametrizando com cobertura por vértices

Ideia:

- Partimos de uma cobertura por vértices X de tamanho k
- Reduzimos para PLI com poucas variáveis ($p = f(k)$)

Parametrizando com cobertura por vértices

Ideia:

- Partimos de uma cobertura por vértices X de tamanho k
- Reduzimos para PLI com poucas variáveis ($p = f(k)$)

Visão geral do algoritmo:

1. Adivinhamos uma ordenação $X = \{u_1, u_2, \ldots, u_k\}$

↳ i.e. enumeramos os $k!$ ordenações de X.

Parametrizando com cobertura por vértices

Ideia:

- Partimos de uma cobertura por vértices X de tamanho k
- Reduzimos para PLI com poucas variáveis ($p = f(k)$)

Visão geral do algoritmo:

1. Adivinhamos uma ordenação $X = \{u_1, u_2, \ldots, u_k\}$
2. Supomos que $\pi(u_1) < \pi(u_2) < \cdots < \pi(u_k)$ na ordenação ótima

Parametrizando com cobertura por vértices

Ideia:

- Partimos de uma cobertura por vértices X de tamanho k
- Reduzimos para PLI com poucas variáveis ($p = f(k)$)

Visão geral do algoritmo:

$K = S$

1. Adivinhamos uma ordenação $X = \{u_1, u_2, \ldots, u_k\}$
2. Supomos que $\pi(u_1) < \pi(u_2) < \cdots < \pi(u_k)$ na ordenação ótima
3. Inserimos $V(G) \setminus X$ "entre" os vértices de X

Parametrizando com cobertura por vértices

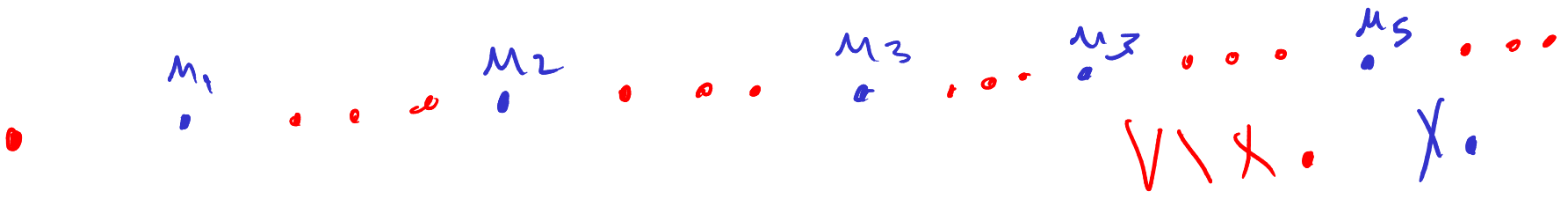
Ideia:

- Partimos de uma cobertura por vértices X de tamanho k
- Reduzimos para PLI com poucas variáveis ($p = f(k)$)

Visão geral do algoritmo:

1. Adivinhamos uma ordenação $X = \{u_1, u_2, \ldots, u_k\}$
2. Supomos que $\pi(u_1) < \pi(u_2) < \cdots < \pi(u_k)$ na ordenação ótima
3. Inserimos $V(G) \setminus X$ "entre" os vértices de X

Notação: $X_i = \{u_1, u_2, \ldots, u_k\}$ são os i primeiros vértices de X



Inserindo vértices na ordenação

Fixe uma ordenação π.

Inserindo vértices na ordenação

Fixe uma ordenação π. Particionamos $V(G) \setminus X$ em $L_0, L_1, \ldots, L_k$ adicionando v:

Inserindo vértices na ordenação

Fixe uma ordenação π. Particionamos $V(G) \setminus X$ em $L_0, L_1, \ldots, L_k$ adicionando v:

1. em L_0 se $\pi(v) < \pi(u_0)$,

Inserindo vértices na ordenação

Fixe uma ordenação π. Particionamos $V(G) \setminus X$ em $L_0, L_1, \ldots, L_k$ adicionando v:

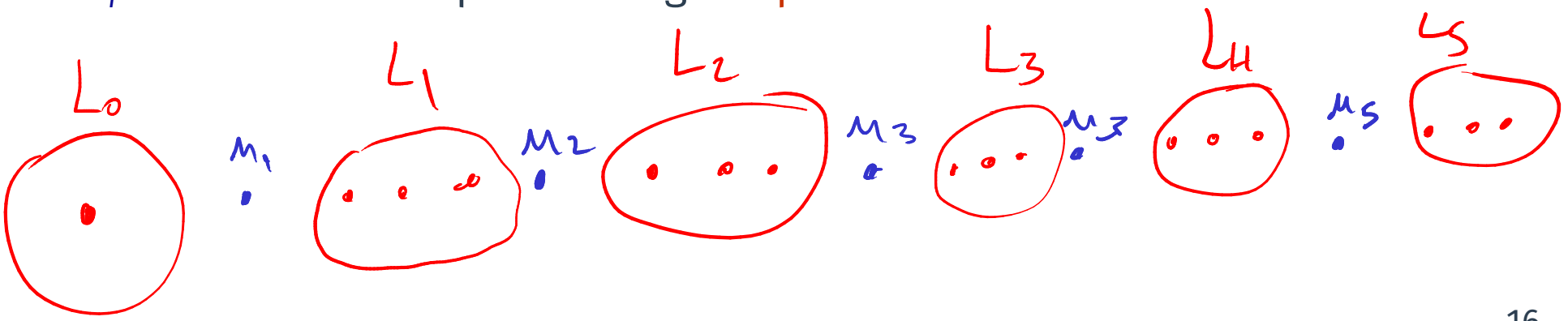
1. em L_0 se $\pi(v) < \pi(u_0)$,
2. em L_i se $\pi(u_i) < \pi(v) < \pi(u_{i+1})$.

Inserindo vértices na ordenação

Fixe uma ordenação π. Particionamos $V(G) \setminus X$ em $L_0, L_1, \ldots, L_k$ adicionando v:

1. em L_0 se $\pi(v) < \pi(u_0)$,
2. em L_i se $\pi(u_i) < \pi(v) < \pi(u_{i_1})$.

L_i são os vértices que vem logo depois de v.

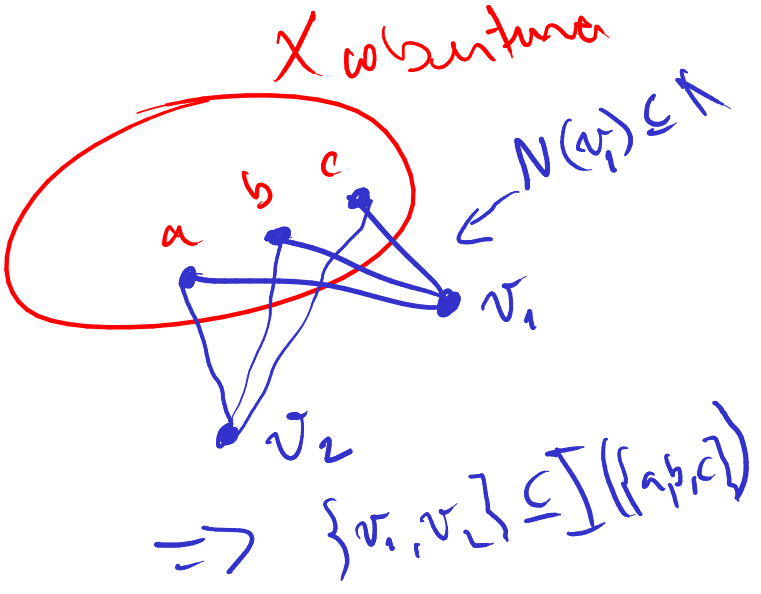


Ordenação interna

Definição

O tipo de um vértice $v \in V(G) \setminus X$ é $S := N(v) \subseteq X$.

- $I(S)$ é o conjunto de vértices com tipo S



Ordenação interna

Definição

O tipo de um vértice $v \in V(G) \setminus X$ é $S := N(v) \subseteq X$.

- $I(S)$ é o conjunto de vértices com tipo S

Fixe L_i:

- Todo vértice de $I(S) \cap L_i$ tem o mesmo custo

Ordenação interna

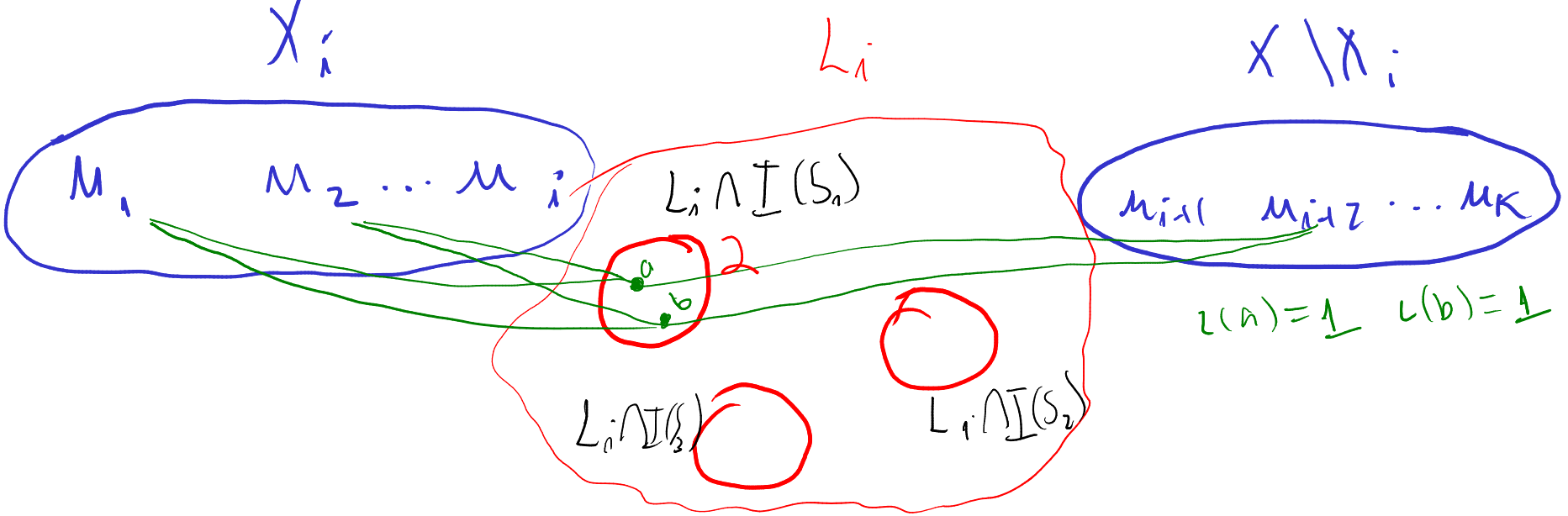
Definição

O tipo de um vértice $v \in V(G) \setminus X$ é $S := N(v) \subseteq X$.

- $I(S)$ é o conjunto de vértices com tipo S

Fixe L_i:

- Todo vértice de $I(S) \cap L_i$ tem o mesmo custo
- A ordem entre os vértices $I(S) \cap L_i$ não altera o valor total



Ordenação interna

Definição

O tipo de um vértice $v \in V(G) \setminus X$ é $S := N(v) \subseteq X$.

- $I(S)$ é o conjunto de vértices com tipo S

Fixe L_i:

- Todo vértice de $I(S) \cap L_i$ tem o mesmo custo
- A ordem entre os vértices $I(S) \cap L_i$ não altera o valor total

  $\Rightarrow$ só definimos quantos elementos de $I(S)$ entram em L_i

Ordenação interna

Definição

O tipo de um vértice $v \in V(G) \setminus X$ é $S := N(v) \subseteq X$.

- $I(S)$ é o conjunto de vértices com tipo S

Fixe L_i:

- Todo vértice de $I(S) \cap L_i$ tem o mesmo custo
- A ordem entre os vértices $I(S) \cap L_i$ não altera o valor total

  $\Rightarrow$ só definimos quantos elementos de $I(S)$ entram em L_i

O custo de um vértice $v \in I(S) \cap L_i$ é

$$z_S^i := ||S \cap X_i| - |S \cap (X \setminus X_i)||$$

$L_\pi(v) = \{v' \in N(v) : \pi(v') < \pi(v)\}$

$= N(v) \cap X_i$

$= X \cap N_i$

Definindo variáveis

Criamos as seguintes variáveis:

Definindo variáveis

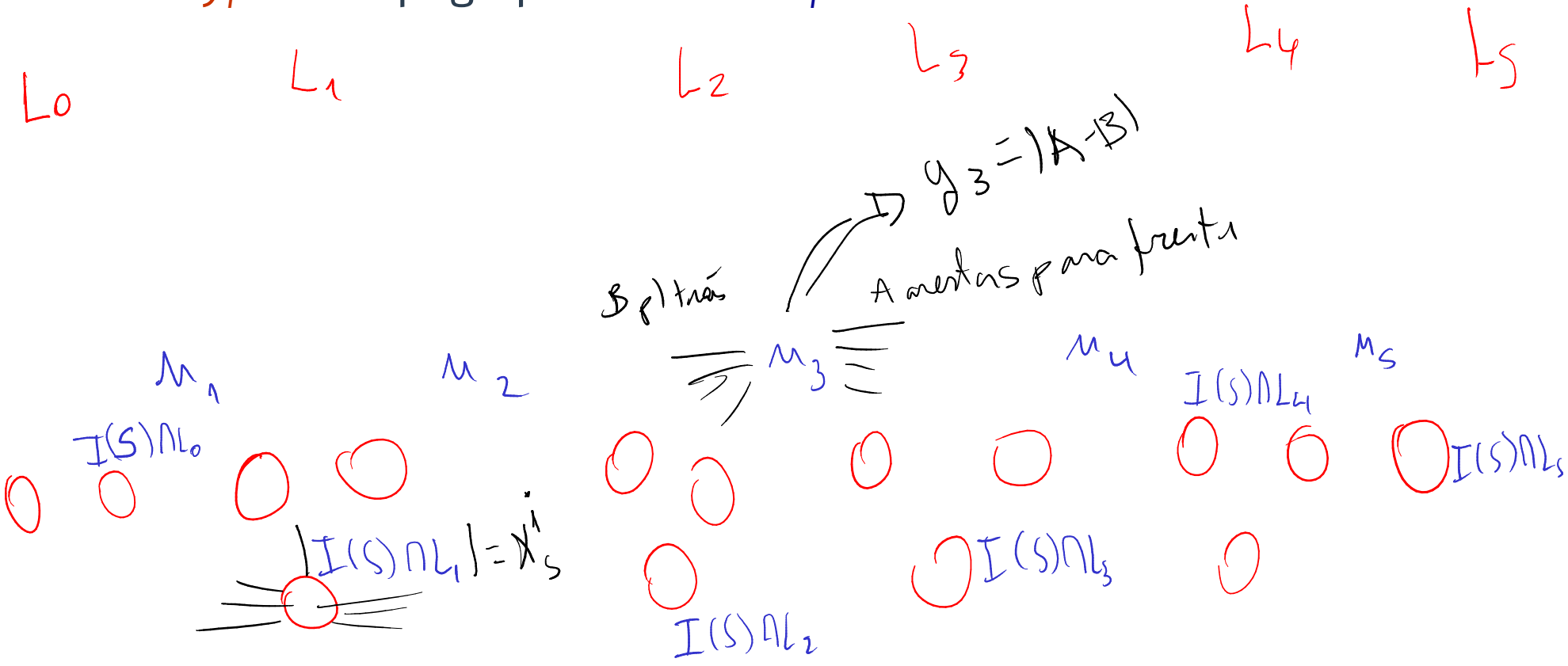
Criamos as seguintes variáveis:

- x_S^i: número de vértices em $I(S) \cap L_i$

Definindo variáveis

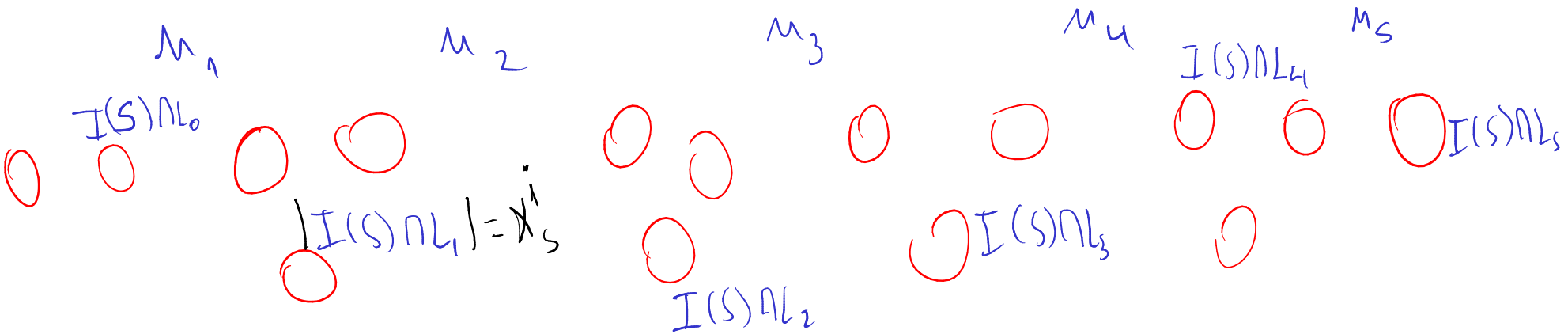
Criamos as seguintes variáveis:

- x_S^i: número de vértices em $I(S) \cap L_i$
- y_i: custo pago pelo vértice u_i



Viabilidade (a)

(a) Inserimos todos os vértices?



. Para todo S, espalhamos $I(S)$ entre $L_0, L_1, L_2, \ldots, L_K$.

$\Rightarrow$ Apenas contamos: $\sum_{i=0}^{K} x_S^i = |I(S)|$.

Viabilidade (a)

(a) Inserimos todos os vértices?

- Verificamos para cada tipo $S \subseteq X$:

Viabilidade (a)

(a) Inserimos todos os vértices?

- Verificamos para cada tipo $S \subseteq X$:

$$\sum_{i=0}^{k} x_S^i = |I(S)|$$

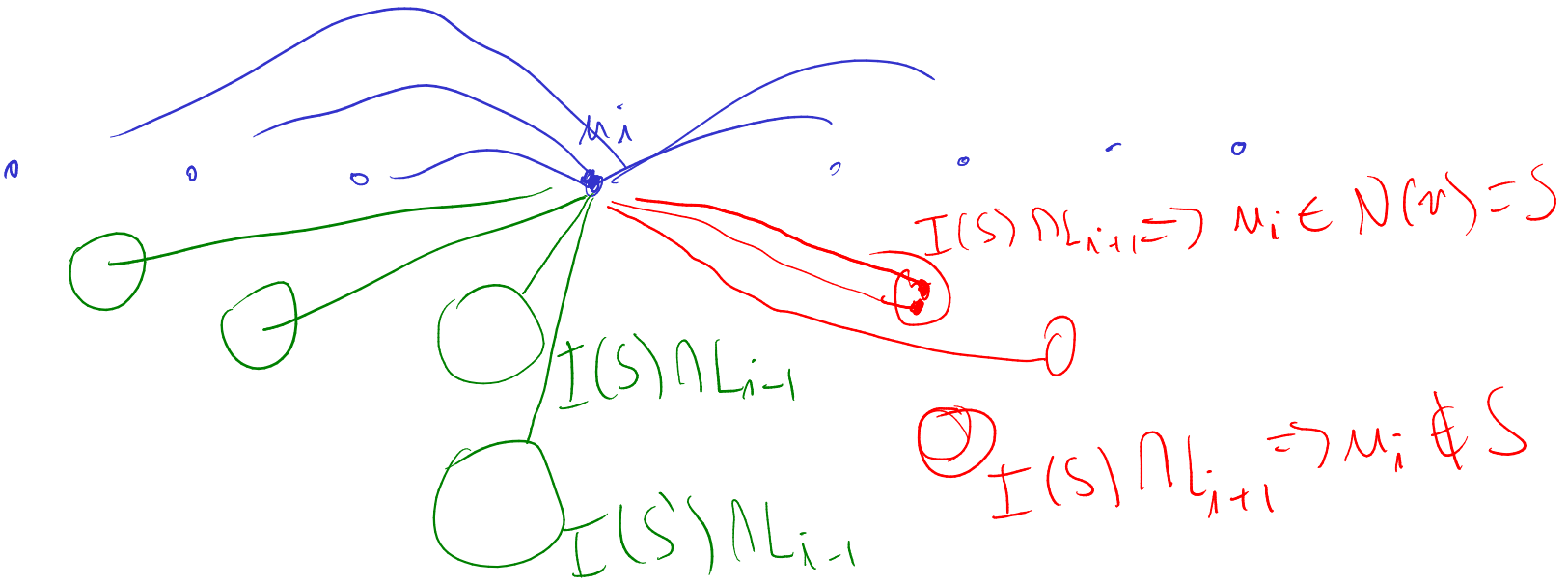
Viabilidade (b)

(b) Calculando o custo de $u_i \in X$

Viabilidade (b)

(b) Calculando o custo de $u_i \in X$

- O desbalanço em X é constante:

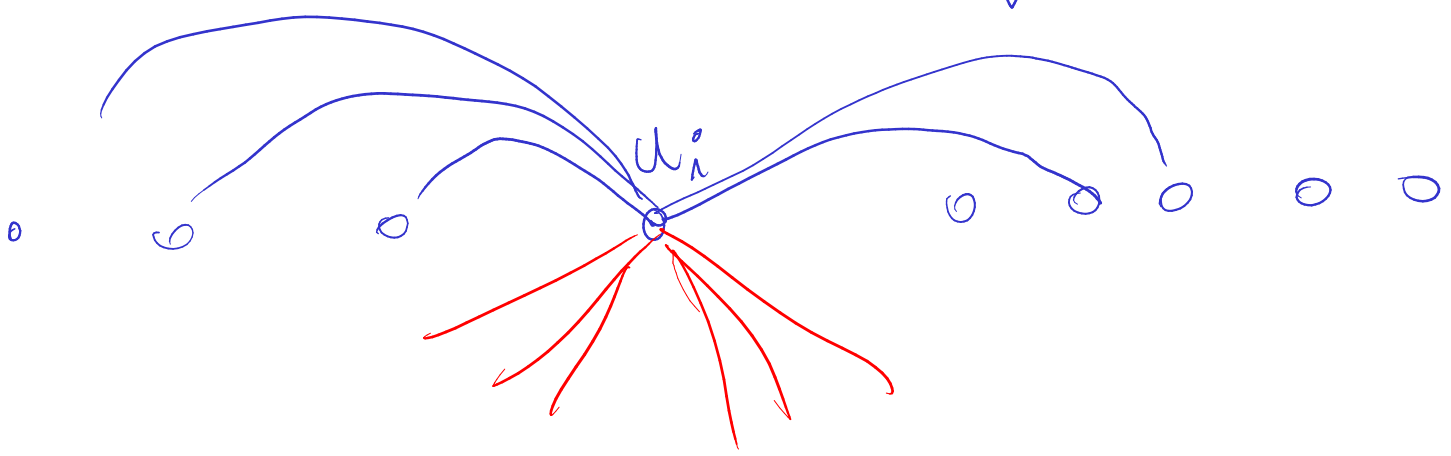


Viabilidade (b)

(b) Calculando o custo de $u_i \in X$

- O desbalanço em X é constante:

$$e_i := |N(u_i) \cap X_{i-1}| - |N(u_i) \cap (X \setminus X_i)|$$



Viabilidade (b)

(b) Calculando o custo de $u_i \in X$

- O desbalanço em X é constante:

$$e_i := |N(u_i) \cap X_{i-1}| - |N(u_i) \cap (X \setminus X_i)|$$

- O desbalanço em $V(G) \setminus X$ é variável:

Viabilidade (b)

(b) Calculando o custo de $u_i \in X$

- O desbalanço em X é constante:

$$e_i := |N(u_i) \cap X_{i-1}| - |N(u_i) \cap (X \setminus X_i)|$$

- O desbalanço em $V(G) \setminus X$ é variável:

$$\sum_{\substack{S \subseteq X \\ u_i \in S}} \left(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j \right)$$

Viabilidade (b)

(b) Calculando o custo de $u_i \in X$

- O desbalanço em X é constante:

$$e_i := |N(u_i) \cap X_{i-1}| - |N(u_i) \cap (X \setminus X_i)|$$

- O desbalanço em $V(G) \setminus X$ é variável:

$$\sum_{\substack{S \subseteq X \\ u_i \in S}} \left(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j \right)$$

- Combinando:

$$y_i \geq \left| e_i + \sum_{\substack{S \subseteq X \\ u_i \in S}} \left(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j \right) \right|$$

Função objetivo

Separamos os custos em:

Função objetivo

Separamos os custos em:

- Custos de X:

Função objetivo

Separamos os custos em:

- Custos de X:

$$\sum_{i=1}^{k} y_i$$

Função objetivo

Separamos os custos em:

- Custos de X:

$$\sum_{i=1}^{k} y_i$$

- Custos de L_i:

Função objetivo

Separamos os custos em:

- Custos de X:

$$\sum_{i=1}^{k} y_i$$

- Custos de L_i:

$$\sum_{i=0}^{k} \sum_{S \subseteq X} z_S^i x_S^i$$

Função objetivo

Separamos os custos em:

- Custos de X:

$$\sum_{i=1}^{k} y_i$$

- Custos de L_i:

$$\sum_{i=0}^{k} \sum_{S \subseteq X} z_S^i x_S^i$$

- Combinando:

$$\sum_{i=1}^{k} y_i + \sum_{i=0}^{k} \sum_{S \subseteq X} z_S^i x_S^i$$

PLI resultante

PLI

$$\min \quad \sum_{i=1}^{k} y_i + \sum_{i=0}^{k} \sum_{S \subseteq X} z_S^i x_S^i$$

$$\text{s.a.} \quad \sum_{i=0}^{k} x_S^i = |I(S)| \qquad \forall S \subseteq X$$

$$y_i \geq \left| e_i + \sum_{\substack{S \subseteq X \\ u_i \in S}} \left(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j \right) \right| \qquad \forall 1 \leq i \leq k$$

$$x_S^i \geq 0 \qquad \forall 0 \leq i \leq k, S \subseteq X$$

partições V\X em I(S)

todo I(S) é contado

já está bem definido

$A \geq |B| \iff A \geq B$ e $A \geq -B$.

PLI resultante

PLI

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{k} y_i + \sum_{i=0}^{k} \sum_{S \subseteq X} z_S^i x_S^i & \\
\text{s.a.} \quad & \sum_{i=0}^{k} x_S^i = |I(S)| & \forall S \subseteq X \\
& y_i \geq \left| e_i + \sum_{\substack{S \subseteq X \\ u_i \in S}} \left(\sum_{j=0}^{i-1} x_S^j - \sum_{j=i}^{k} x_S^j \right) \right| & \forall 1 \leq i \leq k \\
& x_S^i \geq 0 & \forall 0 \leq i \leq k, S \subseteq X
\end{aligned}
$$

Observação: o segundo grupo de desigualdades pode ser convertida em desigualdades lineares

Concluindo

Teorema

*O problema do Desbalanceamento mínimo com cobertura por vértices X, parametrizado por $k = |X|$, é **FPT**.*

Prova:

- Construímos cada um dos $K!$ PLIs em tempo:
  - Temos que se a solução ótima, então $|y_i|, |x_s^i| \leq n$ i.e. $M_1 \leq n$.
  - Temos que o maior coeficiente é $\max\{d_i, z_s^i\} \leq n$.
  - Temos $O(2^K \cdot K)$ variáveis
  - Resolvemos o PLI em tempo FPT.

Teorema de Robertson e Seymour

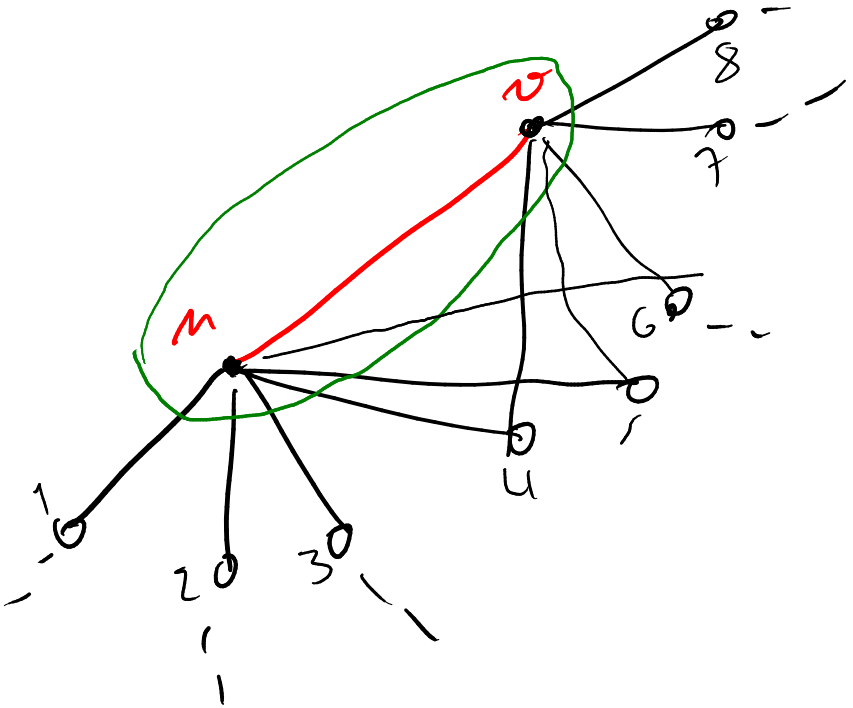
Menores

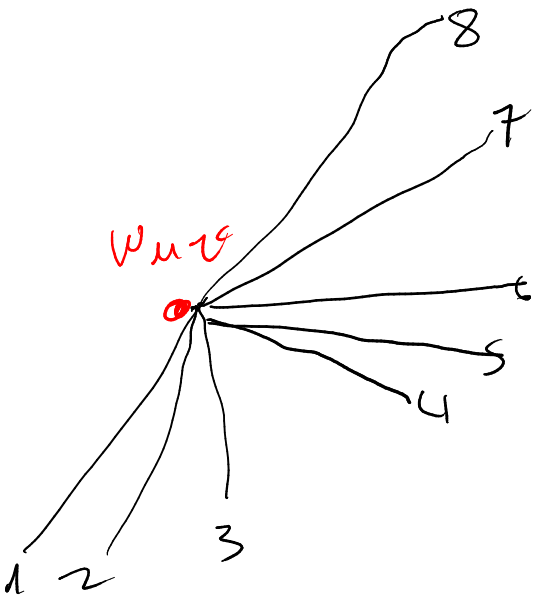
Contração de aresta: Dado um grafo G e uma aresta uv, definimos G/uv como o grafo obtido por:

Menores

Contração de aresta: Dado um grafo G e uma aresta uv, definimos G/uv como o grafo obtido por:

1. Remova u e v de G; e arestas incidentes
2. Adicione um novo vértice w_{uv};
3. Adicione uma aresta de w_{uv} para cada vértice de $N_G(u) \cup N_G(v) \setminus \{u, v\}$.





Menores

Contração de aresta: Dado um grafo G e uma aresta uv, definimos G/uv como o grafo obtido por:

1. Remova u e v de G;
2. Adicione um novo vértice w_{uv};
3. Adicione uma aresta de w_{uv} para cada vértice de $N_G(u) \cup N_G(v) \setminus \{u, v\}$.

Definição

Dizemos que um grafo H é um *minor* (menor) de G, denotado por $H \leq_m G$ se obtemos H a partir de G por meio de:

Menores

Contração de aresta: Dado um grafo G e uma aresta uv, definimos G/uv como o grafo obtido por:

1. Remova u e v de G;
2. Adicione um novo vértice w_{uv};
3. Adicione uma aresta de w_{uv} para cada vértice de $N_G(u) \cup N_G(v) \setminus \{u, v\}$.

Definição

Dizemos que um grafo H é um *minor* (menor) de G, denotado por $H \leq_m G$ se obtemos H a partir de G por meio de:

- remoção de vértices e arestas;

Menores

Contração de aresta: Dado um grafo G e uma aresta uv, definimos G/uv como o grafo obtido por:

1. Remova u e v de G;
2. Adicione um novo vértice w_{uv};
3. Adicione uma aresta de w_{uv} para cada vértice de $N_G(u) \cup N_G(v) \setminus \{u, v\}$.

Definição

Dizemos que um grafo H é um *minor* (menor) de G, denotado por $H \leq_m G$ se obtemos H a partir de G por meio de:

- remoção de vértices e arestas; (obtenho um subgrafo de G)
- contração de arestas. (de subgrafo)

Outra definição

Uma definição equivalente:

Minor model

Um grafo H é um minor de G se para cada $h \in V(H)$ existe $V_h \subseteq V(G)$ e:

Outra definição

Um definição equivalente:

Minor model

Um grafo H é um minor de G se para cada $h \in V(H)$ existe $V_h \subseteq V(G)$ e:

(a) $G[V_h]$ é conexo;

Outra definição

Um definição equivalente:

Minor model

Um grafo H é um minor de G se para cada $h \in V(H)$ existe $V_h \subseteq V(G)$ e:

(a) $G[V_h]$ é conexo;

(b) V_g e V_h são disjuntos para todos $v, h \in V(H)$;

Outra definição

Um definição equivalente:

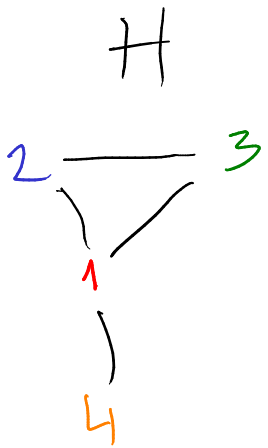
Minor model

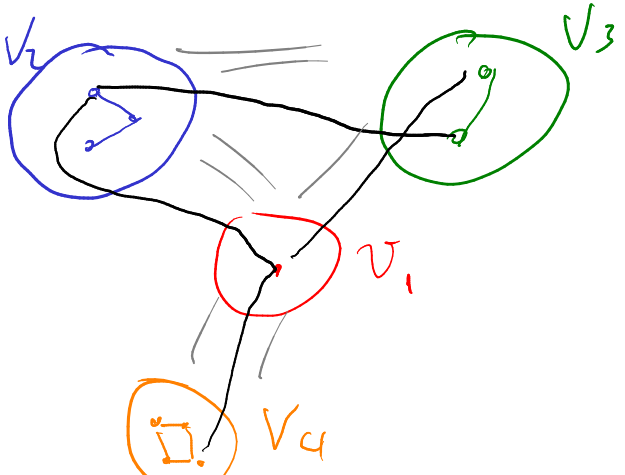
Um grafo H é um minor de G se para cada $h \in V(H)$ existe $V_h \subseteq V(G)$ e:

(a) $G[V_h]$ é conexo;

(b) V_g e V_h são disjuntos para todos $v, h \in V(H)$;

(c) para todo $gh \in H$, existe $v_g \in V_g$ e $v_h \in V_h$ tais que $v_g v_h \in E(G)$.





Famílias fechadas sob minors

Famílias fechadas sob minors

Dizemos que uma família $\mathcal{G}$ é fechada sob a relação de minor se:

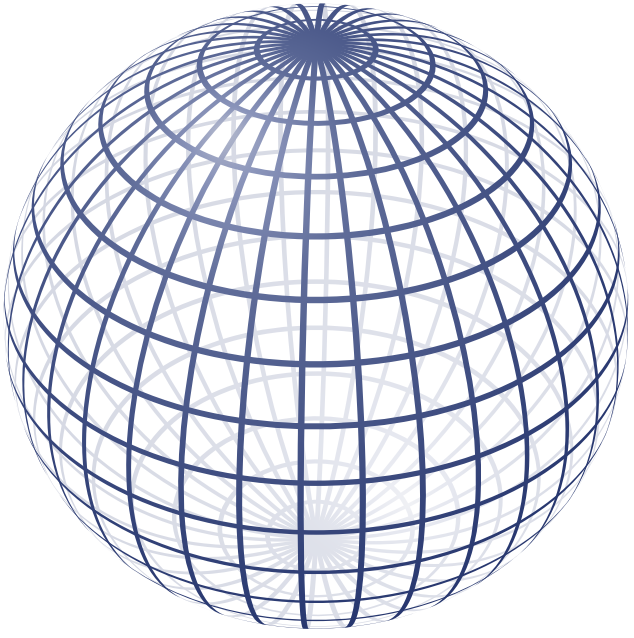
$$G \in \mathcal{G} \text{ e } H \leq_m G \Longrightarrow H \in \mathcal{G}.$$

Exemplos: grafos imersíveis

- Grafo planar: G é planar se puder ser imerso em uma esfera

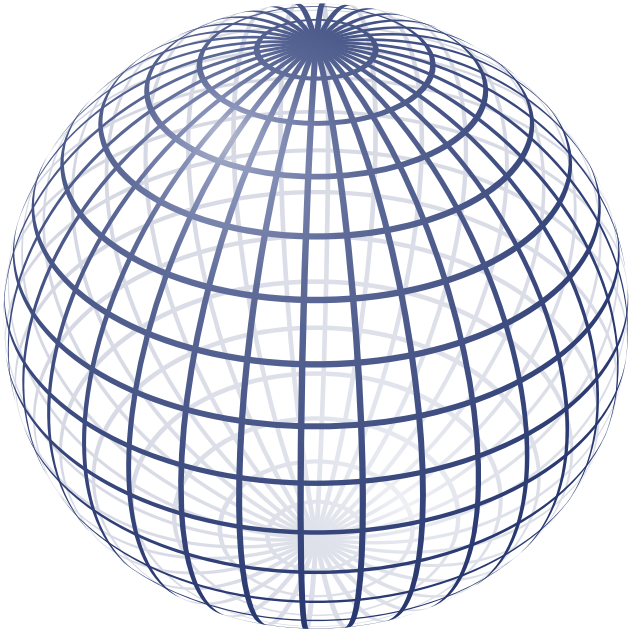
Exemplos: grafos imersíveis

- Grafo planar: G é planar se puder ser imerso em uma esfera



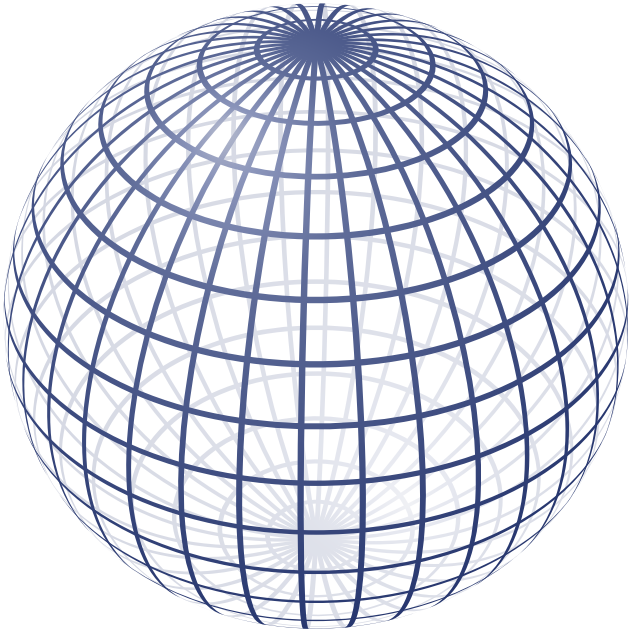
Exemplos: grafos imersíveis

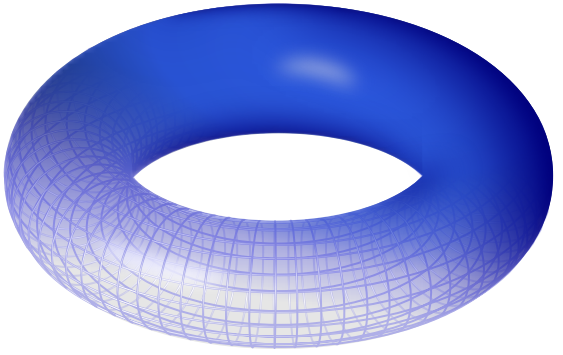
- Grafo planar: G é planar se puder ser imerso em uma esfera
- Grafos toroidal: G é toroidal se puder ser imerso em um toro.



Exemplos: grafos imersíveis

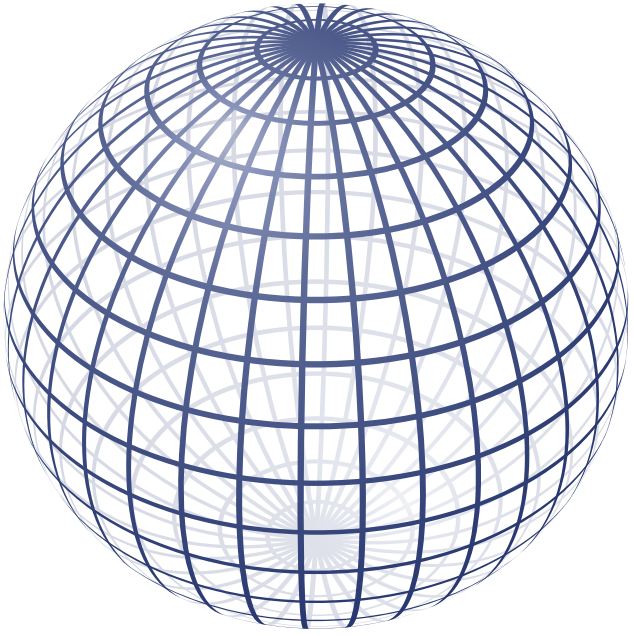
- Grafo planar: G é planar se puder ser imerso em uma esfera
- Grafos toroidal: G é toroidal se puder ser imerso em um toro.

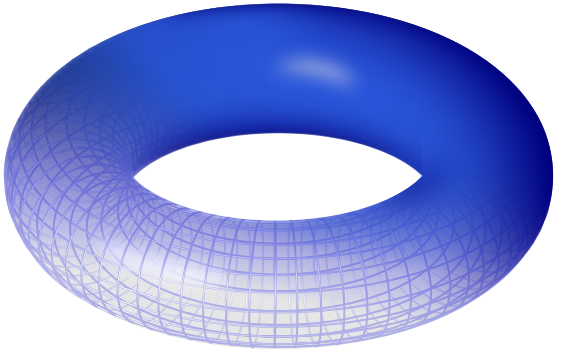




Exemplos: grafos imersíveis

- Grafo planar: G é planar se puder ser imerso em uma esfera
- Grafos toroidal: G é toroidal se puder ser imerso em um toro.





Mais genericamente, grafos com *genus* g.

Exemplos: instâncias sim da cobertura por vértices

Proposição

Seja G um grafo, $X \subseteq V(G)$ uma cobertura por vértices de G e H um minor de G. Então existe uma cobertura por vértices de H com no máximo $|X|$ vértices.

Teorema de Robertson-Seymour

Teorema

A classe de todos os grafos é well-quasi-ordered (“boa” pré-ordem) sob a relação de minor.

Teorema de Robertson-Seymour

Teorema

A classe de todos os grafos é well-quasi-ordered ("boa" pré-ordem) sob a relação de minor.

Ou seja, em qualquer sequência infinita de grafos, existem dois grafos não-isomorfos tal que um é um minor do outro.

Grafos proibidos

Uma consequência do teorema de Robertson-Seymour é que existe uma caracterização natural de cada classe de grafos fechada sob minor baseada em grafos proibidos.

Grafos proibidos

Uma consequência do teorema de Robertson-Seymour é que existe uma caracterização natural de cada classe de grafos fechada sob minor baseada em grafos proibidos.

Corolário

Para cada classe de grafos fechada sob minor $\mathcal{G}$, existe um conjunto finito de grafos $Forb(\mathcal{G})$ tal que para cada grafo G:

Grafos proibidos

Uma consequência do teorema de Robertson-Seymour é que existe uma caracterização natural de cada classe de grafos fechada sob minor baseada em grafos proibidos.

Corolário

Para cada classe de grafos fechada sob minor $\mathcal{G}$, existe um conjunto finito de grafos $Forb(\mathcal{G})$ tal que para cada grafo G:

$$G \in \mathcal{G} \iff \text{não existe } H \in Forb(\mathcal{G}) : H \leq_m G$$

consequência algorítmica

Outro teorema de Robertson-Seymour:

Teorema

Existe uma função computável e um algoritmo que, dados H e G, verifica em tempo $f(H)|V(G)|^3$ se $H \leq_m G$.

consequência algorítmica

Outro teorema de Robertson-Seymour:

Teorema

Existe uma função computável e um algoritmo que, dados H e G, verifica em tempo $f(H)|V(G)|^3$ se $H \leq_m G$.

Teorema

Seja $\mathcal{G}$ é fechada sob minor. Então existe uma contante $c_{\mathcal{G}}$ que depende somente de $\mathcal{G}$, tal que para dodo grafo com n vértices G, verificar se $G \in \mathcal{G}$ pode ser decidido em tempo $c_{\mathcal{G}} \cdot n^3$.

Problemas Tratáveis por Parâmetro fixos não uniformemente

Consequência:

Problemas Tratáveis por Parâmetro fixos não uniformemente

Consequência: e.g., para cobertura por vértices obtemos:

Problemas Tratáveis por Parâmetro fixos não uniformemente

Consequência: e.g., para cobertura por vértices obtemos:

Corolário

Para cada $k \geq 1$ fixo, existe uma constante $c_{\mathcal{G}}$ e um algoritmo que, dado um grafo G com n vértices, verifica em tempo $c_{\mathcal{G}} \cdot n^3$ se G tem uma cobertura por vértices de tamanho no máximo k.

Problemas Tratáveis por Parâmetro fixos não uniformemente

Consequência: e.g., para cobertura por vértices obtemos:

Corolário

Para cada $k \geq 1$ fixo, existe uma constante $c_{\mathcal{G}}$ e um algoritmo que, dado um grafo G com n vértices, verifica em tempo $c_{\mathcal{G}} \cdot n^3$ se G tem uma cobertura por vértices de tamanho no máximo k.

Definição (*Nonuniformly Fixed Parameter Tractable*)

Dizemos que um problema parametrizado Q é tratável por parâmetro fixo não uniformemente se, e somente se, existe uma constante α, uma função $f : \mathbb{N} \to \mathbb{N}$ e uma coleção de algoritmos $(\mathcal{A}_k)_{k \in \mathbb{N}}$ tais que:

Problemas Tratáveis por Parâmetro fixos não uniformemente

Consequência: e.g., para cobertura por vértices obtemos:

Corolário

Para cada $k \geq 1$ fixo, existe uma constante $c_{\mathcal{G}}$ e um algoritmo que, dado um grafo G com n vértices, verifica em tempo $c_{\mathcal{G}} \cdot n^3$ se G tem uma cobertura por vértices de tamanho no máximo k.

Definição (*Nonuniformly Fixed Parameter Tractable*)

Dizemos que um problema parametrizado Q é tratável por parâmetro fixo não uniformemente se, e somente se, existe uma constante α, uma função $f : \mathbb{N} \to \mathbb{N}$ e uma coleção de algoritmos $(\mathcal{A}_k)_{k \in \mathbb{N}}$ tais que:

1. $(x, k) \in P$ sss $A_k(x)$ responde sim;
2. $A_k(x)$ executa em tempo $f(k)|x|^{\alpha}$.

Uma classe de problemas

$\mathcal{G}$-Problema de remoção de vértices

Para uma classe de grafos $\mathcal{G}$, o problema de remoção de vértices é o problema que, dado G e um inteiro k, queremos saber se existe um conjunto $X \subseteq V(G)$ tal que:

Uma classe de problemas

$\mathcal{G}$-Problema de remoção de vértices

Para uma classe de grafos $\mathcal{G}$, o problema de remoção de vértices é o problema que, dado G e um inteiro k, queremos saber se existe um conjunto $X \subseteq V(G)$ tal que:

1. $|X| \leq k$;
2. $G - X \in \mathcal{G}$.

Uma classe de problemas

$\mathcal{G}$-Problema de remoção de vértices

Para uma classe de grafos $\mathcal{G}$, o problema de remoção de vértices é o problema que, dado G e um inteiro k, queremos saber se existe um conjunto $X \subseteq V(G)$ tal que:

1. $|X| \leq k$;
2. $G - X \in \mathcal{G}$.

Teorema

*Se $\mathcal{G}$ é fechado sob minor, então o $\mathcal{G}$-problema de remoção de vértices é não uniformemente **FPT**, quando parametrizado por k.*