Algoritmos Parametrizados

Programação Linear e Ramificação

Lehilton Pedrosa

Segundo Semestre de 2016

Instituto de Computação – Unicamp

Roteiro

1. Programação Linear
2. Núcleo baseado em programação linear
3. Cobertura por vértice (excedendo PL)
4. Transversal para ciclos ímpares
5. Problema da cadeia de caracteres mais próxima

Programação Linear

Programa linear

Um programa linear (PL) é formado por

- uma função linear em variáveis reais; ✓
- conjunto de desigualdades lineares nas variáveis; ✓
- um objetivo maximização ou minimização

MAX
ou
MIN

$$c_1 x_1 + c_2 x_2 + c_3 x_3 + c_n x_n$$

$$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 + a_{1n} x_n \leq b_1$$

Programa linear

Um programa linear (PL) é formado por

- uma função linear em variáveis reais;
- conjunto de desigualdades lineares nas variáveis;
- um objetivo maximização ou minimização

Exemplo:

$$\begin{array}{rl} \text{minimize} & 2x + y \\ \text{sujeito a} & x - 2y \geq 3 \\ & 2x - y \geq 0 \\ & x \geq 0. \end{array}$$

Programação Linear

Programa linear

Um programa linear (PL) é formado por

- uma função linear em variáveis reais;
- conjunto de desigualdades lineares nas variáveis;
- um objetivo maximização ou minimização

Exemplo:

$$\begin{aligned} \text{minimize} \quad & 2x + y \\ \text{sujeito a} \quad & x - 2y \geq 3 \\ & 2x - y \geq 0 \\ & x \geq 0. \end{aligned}$$

Teorema (Khachiyan)

Programação linear é polinomial.

Programação linear inteiro

Programa linear inteiro

Um programa linear inteiro (PLI) é um programa linear com a restrição adicional que variáveis devem ser inteiras.

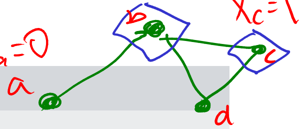
PLI para Cobertura por vértices

PLI é útil para para formular problemas:

PLI para Cobertura por vértices

$x_b = 1$

PLI é útil para para formular problemas:



$x_a = 0$ $x_c = 1$ $x_d = 0$

Cobertura por vértices

$$
\begin{array}{rlll}
\overline{VC}^*(G) = & \text{minimize} & \sum_{v \in V(G)} x_v & \\
& \text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
& & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \\
& & x_v \in \mathbb{Z} & \text{para todo } v \in V(G)
\end{array}
$$

1) Se a cob. por vértices ótima tem tamanho $vc(G) \Rightarrow \overline{VC}^*(G) \leq vc(G)$ e

e existe x viável

t.q. custo de $x = \overline{VC}^*(G)$

2) $\Leftarrow$ volta

PLI para Cobertura por vértices

PLI é útil para para formular problemas:

Cobertura por vértices

$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
& 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \\
& x_v \in \mathbb{Z} & \text{para todo } v \in V(G)
\end{array}
$$

Teorema

Seja $vc(G)$ o valor ótimo do programa acima. Então existe uma cobertura por vértices de tamanho k se, e somente se, $vc(G) \leq k$.

PLI para Cobertura por vértices

PLI é útil para para formular problemas:

Cobertura por vértices

$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
& 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \\
& \cancel{x_v \in \mathbb{Z}} & \cancel{\text{para todo } v \in V(G)}
\end{array}
$$

Teorema

Seja $vc(G)$ o valor ótimo do programa acima. Então existe uma cobertura por vértices de tamanho k se, e somente se, $vc(G) \leq k$.

Corolário

Programação linear inteira é $\mathcal{NP}$-difícil.

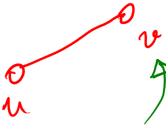
Relaxação

Se removemos as restrições de integralidade:

Relaxação

Se removemos as restrições de integralidade:

$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
 & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G)
\end{array}
$$



$x_v \geq 0{,}7$

$x_u = 0{,}3$

Relaxação

Se removemos as restrições de integralidade:

$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
 & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G)
\end{array}
$$

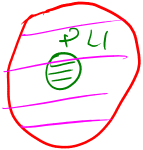
Observações

- interpretamos x_v como a “parte” escolhida de v

Relaxação

Se removemos as restrições de integralidade:

PL



$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
 & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G)
\end{array}
$$

Observações

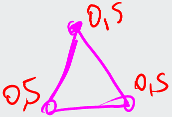
- interpretamos x_v como a "parte" escolhida de v
- se $\mathrm{vc}^*(G)$ é a solução ótimo do PL, então $\mathrm{vc}^*(G) \leq \mathrm{vc}(G)$

Se removemos as restrições de integralidade:

$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
 & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G)
\end{array}
$$

Observações

- interpretamos x_v como a "parte" escolhida de v
- se $\mathrm{vc}^*(G)$ é a solução ótimo do PL, então $\mathrm{vc}^*(G) \leq \mathrm{vc}(G)$
- se G é o triângulo, então $\mathrm{vc}(G) = \frac{3}{2}\underline{\mathrm{vc}^*(G)}$ $\frac{4}{3}$



$2 = vc$ $\frac{3}{2} = vc^*$

Relaxação

Se removemos as restrições de integralidade:

$$
\begin{array}{rll}
\text{minimize} & \sum_{v \in V(G)} x_v & \\
\text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\
 & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G)
\end{array}
$$

Observações

- interpretamos x_v como a "parte" escolhida de v
- se $\mathrm{vc}^*(G)$ é a solução ótimo do PL, então $\mathrm{vc}^*(G) \leq \mathrm{vc}(G)$
- se G é o triângulo, então $\mathrm{vc}(G) = \frac{3}{2}\mathrm{vc}^*(G)$
  - $\rightarrow$ esse fator é chamado de desvio de integralidade de G

Relaxação

Se removemos as restrições de integralidade:

$$\begin{array}{rll} \text{minimize} & \sum_{v \in V(G)} x_v & \\ \text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\ & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \end{array}$$

Observações

- interpretamos x_v como a "parte" escolhida de v
- se $\mathrm{vc}^*(G)$ é a solução ótimo do PL, então $\mathrm{vc}^*(G) \leq \mathrm{vc}(G)$
- se G é o triângulo, então $\mathrm{vc}(G) = \cancel{\tfrac{3}{2}}\,\tfrac{4}{3}\mathrm{vc}^*(G)$
  - → esse fator é chamado de desvio de integralidade de G
  - → o maior desvio de integralidade entre todo G é o desvio de integralidade do PL

Relaxação

Se removemos as restrições de integralidade:

$$\begin{array}{rll} \text{minimize} & \sum_{v \in V(G)} x_v & \\ \text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\ & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \end{array}$$

Observações

- interpretamos x_v como a "parte" escolhida de v
- se $\text{vc}^*(G)$ é a solução ótimo do PL, então $\boxed{\text{vc}^*(G) \leq \text{vc}(G)}$
- se G é o triângulo, então $\text{vc}(G) = \frac{3}{2}\text{vc}^*(G)$
  - → esse fator é chamado de desvio de integralidade de G
  - → o maior desvio de integralidade entre todo G é o desvio de integralidade do PL

Exercício: Mostre que o desvio de integralidade do PL é $2 - \frac{2}{n}$.

Relaxação

Se removemos as restrições de integralidade:

$$\begin{array}{rll} \text{minimize} & \sum_{v \in V(G)} x_v & \\ \text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\ & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \end{array}$$

Observações

- interpretamos x_v como a "parte" escolhida de v
- se $\text{vc}^*(G)$ é a solução ótimo do PL, então $\text{vc}^*(G) \leq \text{vc}(G)$
- se G é o triângulo, então $\text{vc}(G) = \frac{3}{2}\text{vc}^*(G)$
  - → esse fator é chamado de desvio de integralidade de G
  - → o maior desvio de integralidade entre todo G é o desvio de integralidade do PL

Exercício: Mostre que o desvio de integralidade do PL é $2 - \frac{2}{n}$.

Relaxação

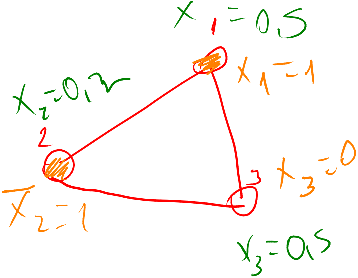
Se removemos as restrições de integralidade:

$$\begin{array}{rll} \text{minimize} & \sum_{v \in V(G)} x_v & \\ \text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\ & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \end{array}$$

Observações

- interpretamos x_v como a "parte" escolhida de v
- se $\mathrm{vc}^*(G)$ é a solução ótimo do PL, então $\mathrm{vc}^*(G) \leq \mathrm{vc}(G)$
- se G é o triângulo, então $\mathrm{vc}(G) = \frac{3}{2}\mathrm{vc}^*(G)$
  - $\rightarrow$ esse fator é chamado de desvio de integralidade de G
  - $\rightarrow$ o maior desvio de integralidade entre todo G é o desvio de integralidade do PL

Exercício: Mostre que o desvio de integralidade do PL é $2 - \frac{2}{n}$.



$$\min \; x_1 + x_2 + x_3$$

$$x_1 + x_2 \geq 1$$
$$x_1 + x_3 \geq 1$$
$$x_2 + x_3 \geq 1$$
$$0 \leq x_i \leq 1$$

$PLI = 2$

$PL = 3/2$

$$GAP = \frac{PLI}{PL} = \frac{2}{3/2} = 4/3$$

$K_n \qquad x_i = 0{,}5 \;\; \forall i \qquad PL = \dfrac{n}{2} \qquad \Rightarrow GAP \simeq 2$

$PLI = n-1$

Relaxação

Se removemos as restrições de integralidade:

$$\begin{array}{rll} \text{minimize} & \sum_{v \in V(G)} x_v & \\ \text{sujeito a} & x_u + x_v \geq 1 & \text{para todo } uv \in E(G) \\ & 0 \leq x_v \leq 1 & \text{para todo } v \in V(G) \end{array}$$

Observações

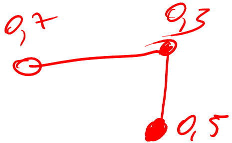
- interpretamos x_v como a "parte" escolhida de v
- se $\mathrm{vc}^*(G)$ é a solução ótimo do PL, então $\mathrm{vc}^*(G) \leq \mathrm{vc}(G)$
- se G é o triângulo, então $\mathrm{vc}(G) = \frac{3}{2}\mathrm{vc}^*(G)$
  - $\rightarrow$ esse fator é chamado de desvio de integralidade de G
  - $\rightarrow$ o maior desvio de integralidade entre todo G é o desvio de integralidade do PL

Exercício: Mostre que o desvio de integralidade do PL é $2 - \frac{2}{n}$.

Estrutura da solução ótima

Seja x uma solução ótima do PL.

Estrutura da solução ótima



Seja x uma solução ótima do PL. Defina:

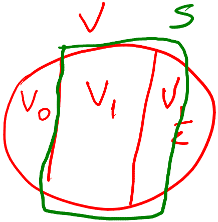
Estrutura da solução ótima

Seja x uma solução ótima do PL. Defina:

- $V_0 = \{v \in V(G) : x_v < \frac{1}{2}\}$;
- $V_{1/2} = \{v \in V(G) : x_v = \frac{1}{2}\}$; ?
- $V_1 = \{v \in V(G) : x_v > \frac{1}{2}\}$;

Estrutura da solução ótima

Seja x uma solução ótima do PL. Defina:



- $V_0 = \{v \in V(G) : x_v < \frac{1}{2}\}$;
- $V_{1/2} = \{v \in V(G) : x_v = \frac{1}{2}\}$;
- $V_1 = \{v \in V(G) : x_v > \frac{1}{2}\}$;

Teorema (Nemhauser-Trotter)

Existe uma cobertura por vértices mínima S de G tal que

$$V_1 \subseteq S \subseteq V_1 \cup V_{1/2}.$$

Seja S^* uma cob mínima.

Defina $S = (S^* \setminus V_0) \cup V_1$

1º) Satisfaz as propriedades:

$$S \cap V_0 = \emptyset \text{ e } V_1 \subseteq S$$

2º) S cobre $E(G)$:

Tome $uv \in E$

→) Se $u \in V_0 \Rightarrow x_u < \frac{1}{2} \Rightarrow x_v > \frac{1}{2}$

$\Rightarrow v \in V_1 \Rightarrow v \in S$.

⋮

3^{o}) $|S| \leq |S^*|$

Por contradição, sup. $|S| > |S^*|$

$$|S| = |S^*| - |S^* \cap V_0| + |V_1 \setminus S^*|$$

$$\frac{|S^*| < |S|}{|V_0 \cap S^*| < |V_1 \setminus S^*|}$$

Seja x a sol. do PL. Vou mostrar $x\overline{M}$ é ótimo.

$$\varepsilon = \min\{|x_v - \tfrac{1}{2}| : v \in V_0 \cup V_1\} > 0$$

Defina

$$y_v = \begin{cases} x_v - \varepsilon, & v \in V_1 \setminus S^* \\ x_v + \varepsilon, & v \in V_0 \cap S^* \\ x_v, & \text{c.c} \end{cases}$$

y é viável: $\forall uv \in E.\,(u \in V_0 \Rightarrow v \in V_1)$

$y_u + y_v = x_u + x_v \geq 1$

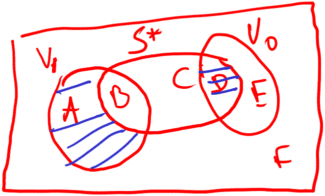
- se $u, v \in V_{1/2}$

$y_u + y_v = x_u + x_v \geq 1$

$-\varepsilon|A| + \varepsilon|D| =$

$\varepsilon(|D| - |A|) < 0$

V



$$\sum_{v \in V} y_v - \sum_{v \in V} x_v < 0$$

$$\sum_{v \in V} y_v - \sum_{v \in V} x_v = \sum_{v \in A} y_v + \sum_{v \in D} y_v - \sum_{v \in A} x_v - \sum_{v \in D} x_v$$

$$= \sum_{v \in A} (\cancel{x_v} - \varepsilon) + \sum_{v \in D} (\cancel{x_v} + \varepsilon) - \cancel{\sum_{v \in A} x_v} - \cancel{\sum_{v \in D} x_v} =$$

Núcleo baseado em programação linear

Observamos que o PL resolve a parte “fácil” do problema:

$G - V_1 \cup V_0$

$k - |V_1|$

Redução baseada em PL

Observamos que o PL resolve a parte "fácil" do problema:

Redução VC.4: Seja x uma solução ótima do PL para instância (G, k):

Redução baseada em PL

Observamos que o PL resolve a parte "fácil" do problema:

Redução VC.4: Seja x uma solução ótima do PL para instância (G, k):

- se $\sum_{v \in V(G)} x_v > k$, então devolva não;

$k < VC^*(G) \leq \underline{VC}(G)$

Redução baseada em PL

Observamos que o PL resolve a parte “fácil” do problema:

Redução VC.4: Seja x uma solução ótima do PL para instância (G, k):

- se $\sum_{v \in V(G)} x_v > k$, então devolva não;
- senão, devolva $(G - V_0 - V_1, k - |V_1|)$.

Redução baseada em PL

Observamos que o PL resolve a parte "fácil" do problema:

Redução VC.4: Seja x uma solução ótima do PL para instância (G, k):

- se $\sum_{v \in V(G)} x_v > k$, então devolva não;
- senão, devolva $(G - V_0 - V_1, k - |V_1|)$.

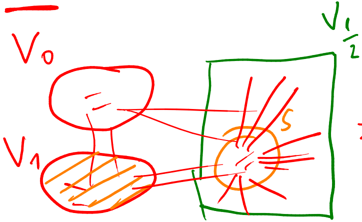
Lema

A redução VC.4 é segura.

Sup. q (G, K) seja sim.

$\Rightarrow vc^*(G) \leq K \Rightarrow$ Passo 1 é seguro ✓

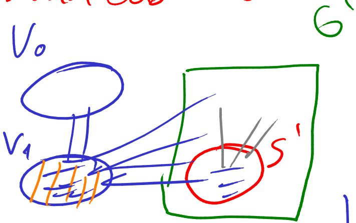
$\Rightarrow$. Seja x uma sol do PL e S uma cob. mín



$\Rightarrow |S \cap V_{\frac{1}{2}}| = |S| - |V_1|$

$= K - |V_1|$

$\Leftarrow$ Seja S' uma cob. de G' de tam. $K' = K - |V_1|$



$u, v \in V_0$

$$x_u + x_v < \frac{1}{2} + \frac{1}{2} = 1 \quad \lightning$$

$u \in V_0$
$v \in V_{1/2}$

$$x_u + x_v < \frac{1}{2} + \frac{1}{2} \quad \lightning$$

$\Rightarrow |S' \cup V_1| = K$ e cobre V. $\square$

Núcleo baseado em PL

Teorema

Cobertura por vértices admite um núcleo com $2k$ vértices.

Aplicamos VC.4:

a) $vc^*(G) > K$, então OK

b) $vc^*(G) \leq K$. Temos:

$$|V(G')| = |V_{\frac{1}{2}}| = 2 \cdot \sum_{v \in V_{\frac{1}{2}}} x_v \leq 2 \sum_{v \in V} x_v \leq 2K. \square$$

Resolvendo o PL usando emparelhamento

Lema

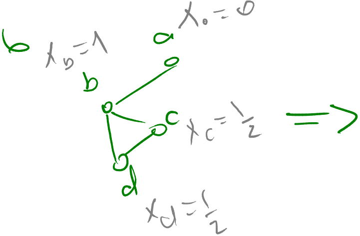
Seja G um grafo com n vértices e m arestas. Uma solução de PL pode ser encontrada em tempo $\mathcal{O}(m\sqrt{n})$.

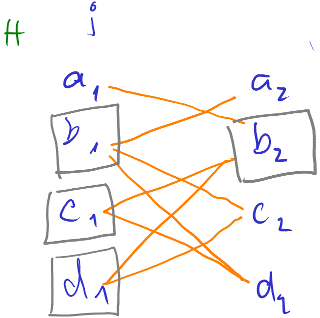
$V = \{v, u, \ldots\}$

$V_1 = \{v_1, u_1, \ldots\}$ $V_2 = \{v_2, u_2, \ldots\}$

Algoritmo:

1. Crie cópias V_1 e V_2 de $V(G)$
2. Crie o grafo bipar- $H = ((V_1, V_2), E_H)$ com arestas:
   - → Para cada $v_1 \in V_1$, $u_2 \in V_2$ adicione $v_1u_2 \in E_H \Leftrightarrow vu \in E(G)$.





3. Use Hopcroft e obtenha uma cobertura por vértices mínima S ($\Theta(m\sqrt{n})$)

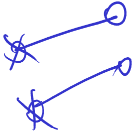
4. Defina:

$$x_v = \begin{cases} 1, & v_1 \text{ e } v_2 \in S \\ 1/2, & v_1 \text{ ou } v_2 \in S \\ 0, & \text{c.c.} \end{cases}$$

5. Devolva x.

(A) x é minimal

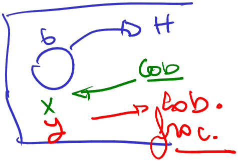
- claro $0 \leq x_v \leq 1 \quad \forall v$
- Seja $uv \in E(G)$:



$$\Rightarrow \{(u_1 v_2), (v_1 u_2)\}$$

$$\Rightarrow \{u_1, v_2\} \cap S \neq \emptyset \text{ e } \{u_2, v_1\} \cap S \neq \emptyset$$

$$\Rightarrow x_u + x_v \geq 1$$



(B) x é ótima do PL

- Seja y ótima do PL.
- defino

$$w(v_i) = y_v \quad \forall v \in V$$

$\forall uv \in E$

$$w(u_1) + w(v_2) = y_u + y_v \geq 1$$

$$\bullet \; vc^*(G) = \sum_{v \in V} y_v = \frac{1}{2} \sum_{v \in V} w(v_1) + w(v_2)$$

$$= \frac{1}{2} \sum_{v_i \in V(H)} w(v_i) \geq \frac{1}{2} vc^*(H)$$

FATO: $vc(H) = vc^*(H)$ pois H é bipart.

$$\underline{vc^*(G)} \geq \frac{1}{2} vc^*(H) = \frac{1}{2} vc(H) = \frac{1}{2} |S|$$

$$= \frac{1}{2} \sum_{v \in V} |\{v_1, v_2\} \cap S| = \sum_{v \in V} x_v$$

$\square$

Consequências

Corolário

Cobertura por vértices admite um núcleo com $2k$ vértices que pode ser computado em tempo $\mathcal{O}(m\sqrt{n})$.

Consequências

Corolário

Cobertura por vértices admite um núcleo com $2k$ vértices que pode ser computado em tempo $\mathcal{O}(m\sqrt{n})$.

Corolário

Existe uma solução ótima do PL tal que $x_v \in \{0, \frac{1}{2}, 1\}$ para todo v e essa solução pode ser encontrada em tempo $\mathcal{O}(m\sqrt{n})$

Consequências

Corolário

Cobertura por vértices admite um núcleo com $2k$ *vértices que pode ser computado em tempo* $\mathcal{O}(m\sqrt{n})$.

Corolário

Existe uma solução ótima do PL tal que $x_v \in \{0, \frac{1}{2}, 1\}$ *para todo* v *e essa solução pode ser encontrada em tempo* $\mathcal{O}(m\sqrt{n})$

A solução acima é chamada de semi-integral.

Cobertura por vértice (excedendo PL)

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom
2. se k é grande: núcleo não adianta muito

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom
2. se k é grande: núcleo não adianta muito

Fatos

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom
2. se k é grande: núcleo não adianta muito

Fatos

- Para instância sim, $\mathrm{vc}(G) \geq \mathrm{vc}^*(G)$

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom
2. se k é grande: núcleo não adianta muito

Fatos

- Para instância sim, $\text{vc}(G) \geq \text{vc}^*(G)$
- Podemos calcular $\text{vc}^*(G)$ em tempo polinomial

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom
2. se k é grande: núcleo não adianta muito

Fatos

- Para instância sim, $\mathrm{vc}(G) \geq \mathrm{vc}^*(G)$
- Podemos calcular $\mathrm{vc}^*(G)$ em tempo polinomial

Parâmetro alternativo: $\mu(G, k) := k - \mathrm{vc}^*(G)$

Mudando o parâmetro

Situação atual: núcleo com $2k$ vértices

Analisando:

1. se k é pequeno: núcleo é muito bom
2. se k é grande: núcleo não adianta muito

Fatos

- Para instância sim, $\mathrm{vc}(G) \geq \mathrm{vc}^*(G)$
- Podemos calcular $\mathrm{vc}^*(G)$ em tempo polinomial

Parâmetro alternativo: $\mu(G, k) := k - \mathrm{vc}^*(G)$

$\rightarrow$ o que excede ou está acima do PL.

Revisitando nossa redução

Lema

Suponha que obtemos (G', k') *após aplicar* VC.4 *sobre* (G, k) *(utilizando uma solução semi-integral* x*). Então* $vc^*(G) - vc^*(G') = vc(G) - vc(G') = V_1^x = k - k'$.

Revisitando nossa redução

Lema

Suponha que obtemos (G', k') *após aplicar* VC.4 *sobre* (G, k) *(utilizando uma solução semi-integral* x*). Então* $vc^*(G) - vc^*(G') = vc(G) - vc(G') = V_1^x = k - k'$.

Consequência: $\mu(G', k') = \mu(G, k)$.

Reduzindo iterativamente

Lema

Dado G, existe um algoritmo que em tempo $\mathcal{O}(mn^{3/2})$:

Reduzindo iterativamente

Lema

Dado G, existe um algoritmo que em tempo $\mathcal{O}(mn^{3/2})$:

- *encontra solução semi-integral de $PL(G)$ que não é toda $1/2$;*

Reduzindo iterativamente

Lema

Dado G, existe um algoritmo que em tempo $\mathcal{O}(mn^{3/2})$:

- *encontra solução semi-integral de $PL(G)$ que não é toda $1/2$;*
- *conclui que $x_v = 1/2$ para $v \in V(G)$ é a única solução de $PL(G)$.*

Reduzindo iterativamente

Lema

Dado G, existe um algoritmo que em tempo $\mathcal{O}(mn^{3/2})$:

- *encontra solução semi-integral de $PL(G)$ que não é toda $1/2$;*
- *conclui que $x_v = 1/2$ para $v \in V(G)$ é a única solução de $PL(G)$.*

Redução VC.5: Se existe solução que não é $x_v = 1/2$ para $v \in V(G)$, então aplique VC.4.

Algoritmo de ramificação

Teorema

Existe um algoritmo para Cobertura por vértices (excedendo PL) com tempo $4^{k-vc^*(G)} \cdot n^{\mathcal{O}(1)}$.

Usando emparelhamento

Observações

Seja M um emparelhamento máximo de G:

- $|M|$ é um limitante inferior da cobertura

Usando emparelhamento

Observações

Seja M um emparelhamento máximo de G:

- $|M|$ é um limitante inferior da cobertura
- $\text{vc}^*(G) \geq |M|$ (por dualidade)

Usando emparelhamento

Observações

Seja M um emparelhamento máximo de G:

- $|M|$ é um limitante inferior da cobertura
- $\mathrm{vc}^*(G) \geq |M|$ (por dualidade)

  $\Rightarrow\ k - \mathrm{vc}^*(G) \leq k - |M|$

Usando emparelhamento

Observações

Seja M um emparelhamento máximo de G:

- $|M|$ é um limitante inferior da cobertura
- $\text{vc}^*(G) \geq |M|$ (por dualidade)
  $\Rightarrow\ k - \text{vc}^*(G) \leq k - |M|$

Teorema

Existe um algoritmo para Cobertura por vértices (excedendo emparelhamento) com tempo $4^{k-\text{vc}^*(G)} \cdot n^{\mathcal{O}(1)}$.

Transversal para ciclos ímpares

Transversal para ciclos ímpares

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

Transversal para ciclos ímpares

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

Problema da transversal para ciclos ímpares mínima

Dado G um grafo não direcionado e um inteiro k.

Transversal para ciclos ímpares

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

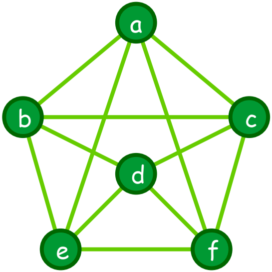
Problema da transversal para ciclos ímpares mínima

Dado G um grafo não direcionado e um inteiro k. Existe uma transversal para ciclos ímpares de tamanho k?

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

Problema da transversal para ciclos ímpares mínima

Dado G um grafo não direcionado e um inteiro k. Existe uma transversal para ciclos ímpares de tamanho k?

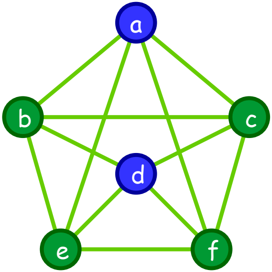


Transversal para ciclos ímpares

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

Problema da transversal para ciclos ímpares mínima

Dado G um grafo não direcionado e um inteiro k. Existe uma transversal para ciclos ímpares de tamanho k?

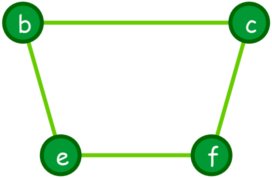


Transversal para ciclos ímpares

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

Problema da transversal para ciclos ímpares mínima

Dado G um grafo não direcionado e um inteiro k. Existe uma transversal para ciclos ímpares de tamanho k?

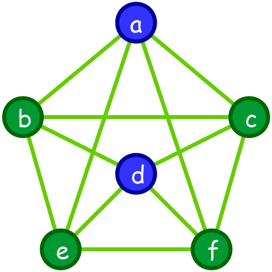


Transversal para ciclos ímpares

Dado G, um conjunto de vértices X é uma transversal para ciclos ímpares se $G - X$ é um grafo bipartido.

Problema da transversal para ciclos ímpares mínima

Dado G um grafo não direcionado e um inteiro k. Existe uma transversal para ciclos ímpares de tamanho k?



Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$

Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$
- $u_1 v_2 \in E(\tilde{G})$ para toda aresta $uv \in E(G)$

Redução para Cobertura por vértices

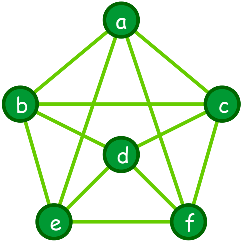
Crie um grafo $\tilde{G}$ tal que:

- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$
- $u_1 v_2 \in E(\tilde{G})$ para toda aresta $uv \in E(G)$
- $u_1 u_2 \in E(\tilde{G})$ para todo vértice $u \in V(G)$

Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

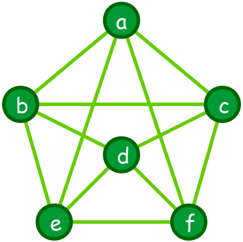
- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$
- $u_1 v_2 \in E(\tilde{G})$ para toda aresta $uv \in E(G)$
- $u_1 u_2 \in E(\tilde{G})$ para todo vértice $u \in V(G)$

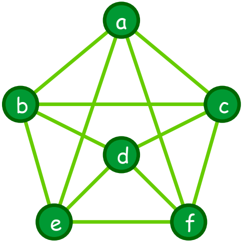


Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$
- $u_1v_2 \in E(\tilde{G})$ para toda aresta $uv \in E(G)$
- $u_1u_2 \in E(\tilde{G})$ para todo vértice $u \in V(G)$

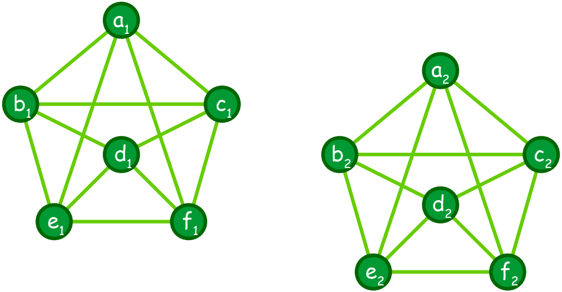




Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

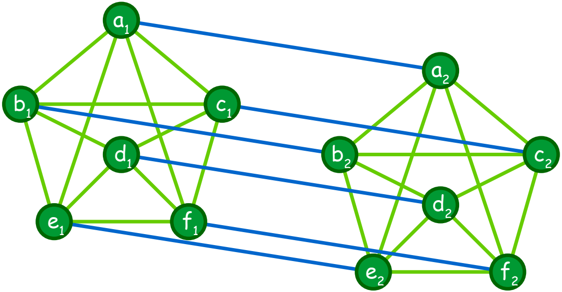
- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$
- $u_1 v_2 \in E(\tilde{G})$ para toda aresta $uv \in E(G)$
- $u_1 u_2 \in E(\tilde{G})$ para todo vértice $u \in V(G)$



Redução para Cobertura por vértices

Crie um grafo $\tilde{G}$ tal que:

- $\tilde{G}$ contém duas cópias de $V(G)$: $V(\tilde{G}) = V_1 \cup V_2$
- $u_1v_2 \in E(\tilde{G})$ para toda aresta $uv \in E(G)$
- $u_1u_2 \in E(\tilde{G})$ para todo vértice $u \in V(G)$



Redução (cont.)

Lema

G tem uma transversal para ciclos ímpares de tamanho k sss $\tilde{G}$ tem uma cobertura por vértices de tamanho $k + n$, onde $n = |V(G)|$.

FPT para Transversal para ciclos

Teorema

Transversal para ciclos ímpares pode ser resolvida em tempo $4^k n^{\mathcal{O}(1)}$.

Problema da cadeia de caracteres mais próxima

Problema da cadeia mais próxima

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

$$x = \text{bananada}$$
$$y = \text{camarada}$$

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

$x =$ bananada

$y =$ camarada L:= |y| = |x| = 8

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

$x =$ bananada $\quad d_H(x,y) = 3$

$y =$ camarada $\quad L := |y| = |x| = 8$

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

x = bananada $\quad d_H(x,y) = 3$

y = camarada $\quad L := |y| = |x| = 8$

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

x = bananada $d_H(x,y) = 3$

y = camarada L:= |y| = |x| = 8

Problema da cadeia de caracteres mais próxima

Dado um conjunto de k cadeias de caracteres $x_1, \ldots, x_k$ com tamanho L e um inteiro d.

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x,y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

x = bananada $\quad d_H(x,y) = 3$

y = camarada $\quad L := |y| = |x| = 8$

Problema da cadeia de caracteres mais próxima

Dado um conjunto de k cadeias de caracteres $x_1, \ldots, x_k$ com tamanho L e um inteiro d. Existe uma cadeia de caracteres y, de tamanho L, tal que $d_H(x_i, y) \leq d$ para todo $1 \leq i \leq k$?

Problema da cadeia mais próxima

Dadas duas cadeias de caracteres x e y de tamanho L, a distância de Hamming, $d_H(x, y)$ entre x e y é a quantidade de índices em que $x[j] \neq y[j]$.

x = bananada $d_H(x,y) = 3$

y = camarada L:= |y| = |x| = 8

Problema da cadeia de caracteres mais próxima

Dado um conjunto de k cadeias de caracteres $x_1, \ldots, x_k$ com tamanho L e um inteiro d. Existe uma cadeia de caracteres y, de tamanho L, tal que $d_H(x_i, y) \leq d$ para todo $1 \leq i \leq k$?

Uma tal cadeia de caracteres y é chamada de centro.

Exemplo de instância: cadeias

$x =$ bananada

$y =$ camarada

$z =$ rabanada

$w =$ laminada

$v =$ laminado

Exemplo de instância: cadeias

$x =$ bananada $d_H(x, A) = 2$

$y =$ camarada $d_H(y, A) = 1$

$z =$ rabanada $d_H(z, A) = 2$

$w =$ laminada $d_H(w, A) = 2$

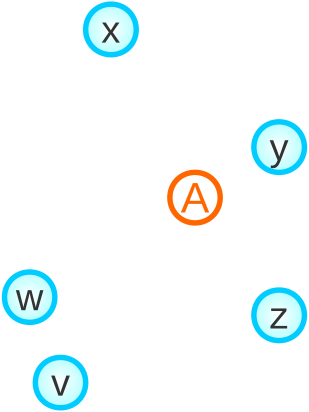
$v =$ laminado $d_H(v, A) = 3$

$A =$ camanada

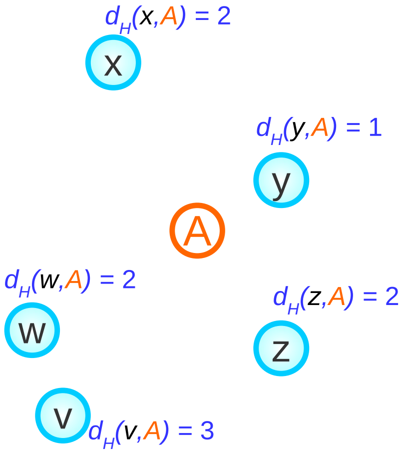
Exemplo de instância: centro

Exemplo de instância: centro

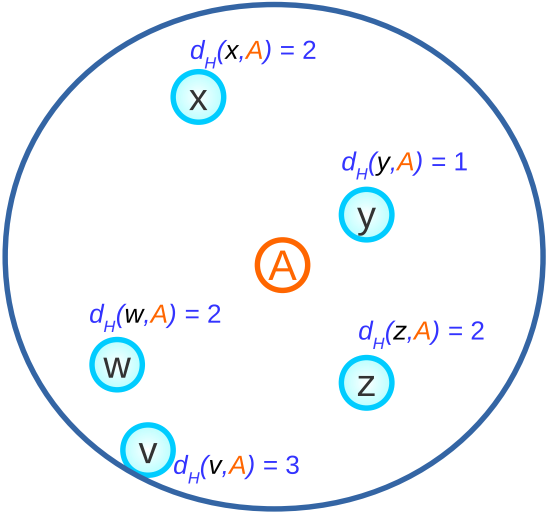
Exemplo de instância: centro



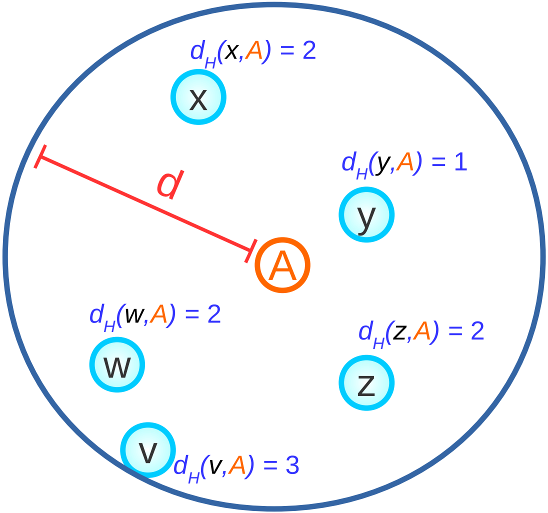
Exemplo de instância: centro



Exemplo de instância: centro



Exemplo de instância: centro



Matriz de caracteres

Representamos o conjunto de cadeias como uma matriz de caracteres.

Matriz de caracteres

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

Representamos o conjunto de cadeias como uma matriz de caracteres.

Matriz de caracteres

coluna 3

```
        b a n a n a d a
linha 2 c a m a r a d a
        r a b a n a d a
        l a m i n a d a
        l a m i n a d o
```

Representamos o conjunto de cadeias como uma matriz de caracteres.

Identificando a parte fácil

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

Identificando a parte fácil

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

- colunas boas são colunas com um único caracteres

Identificando a parte fácil

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

- colunas boas são colunas com um único caracteres

Identificando a parte fácil

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

- colunas boas são colunas com um único caracteres

Identificando a parte fácil

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

- colunas boas são colunas com um único caracteres

Identificando a parte fácil

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

coluna ruim

- colunas boas são colunas com um único caracteres
- colunas ruim são colunas com caracteres diferentes

Identificando a parte fácil

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

coluna ruim

- colunas boas são colunas com um único caracteres
- colunas ruim são colunas com caracteres diferentes

Removendo colunas boas

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

coluna ruim

Redução CS.1: Remova todas colunas boas.

Removendo colunas boas

coluna boa

```
b a n a n a d a
c a m a r a d a
r a b a n a d a
l a m i n a d a
l a m i n a d o
```

coluna ruim

Redução CS.1: Remova todas colunas boas.

Removendo colunas boas

coluna boa

b		n	a	n			a
c		m	a	r			a
r		b	a	n			a
l		m	i	n			a
l		m	i	n			o

coluna ruim

Redução CS.1: Remova todas colunas boas.

Removendo colunas boas

```
b n a n a
c m a r a
r b a n a
l m i n a
l m i n o
```

Redução CS.1: Remova todas colunas boas.

Contando colunas ruins

```
b n a n a
c m a r a
r b a n a
l m i n a
l m i n o
```

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

```
b n a n a
c m a r a
r b a n a
l m i n a
l m i n o
```

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

```
b n a n a
c m a r a
r b a n a
l m i n a
l m i n o
```

camanada

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

```
b n a n a
c m a r a
r b a n a
l m i n a
l m i n o
c m a n a
```

camanada

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

```
c      b n a n a
m      c m a r a
a      r b a n a
n      l m i n a
a      l m i n o
       c m a n a
```

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

c		**b**	n	a	n	a
m		c	m	a	r	a
a		r	b	a	n	a
n		l	m	i	n	a
a		l	m	i	n	o
		c	m	a	n	a

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

c
m
a
n
a

b n a n a
c m a r a
r b a n a
l m i n a
l m i n o
c m a n a

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

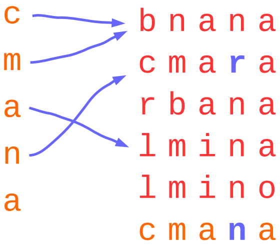
Contando colunas ruins

```
c        b n a n a
m        c m a r a
a        r b a n a
n        l m i n a
a        l m i n o
         c m a n a
```

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins



Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Contando colunas ruins

c

m

a

n

a

b n a n a

c m a r a

r b a n a

l m i n a

l m i n **o**

c m a n **a**

Lema

Seja uma instância sim para o Problema de cadeia de caracteres mais próxima. Então o número de colunas ruins é no máximo $k \times L$.

Algoritmo de ramificação para Cadeia mais próxima

Teorema

O problema da cadeia de caracteres mais próxima pode ser resolvido em tempo $\mathcal{O}(kL + kd(d+1)^d)$.