Disciplinas CCM-001 / MCTA003-17
Análise de Algoritmos (e Estruturas de Dados)

Complexidade computacional

Profa. Carla Negri Lintzmayer

carla.negri@ufabc.edu.br
professor.ufabc.edu.br/~carla.negri

Centro de Matemática, Computação e Cognição – Universidade Federal do ABC

Conteúdo

Estes slides contêm uma primeira parte com foco em redução (Seção 1) e uma segunda parte com uma introdução à complexidade computacional (Seção 2).

Redução como ferramenta para resolver problemas

Redução informalmente

- Problema P: I_P é instância e S_P é solução.

Redução informalmente

- Problema P: I_P é instância e S_P é solução.
- Reduzir um problema A para um problema B significa usar um algoritmo de B para resolver A:

Redução informalmente

- Problema P: I_P é instância e S_P é solução.
- Reduzir um problema A para um problema B significa usar um algoritmo de B para resolver A:

$$I_B \longrightarrow \boxed{ALG_B} \longrightarrow S_B$$

Redução informalmente

- Problema P: I_P é instância e S_P é solução.
- Reduzir um problema A para um problema B significa usar um algoritmo de B para resolver A:

$$I_B \longrightarrow \boxed{ALG_B} \longrightarrow S_B$$

$$I_A \longrightarrow \boxed{\xrightarrow{f} I_B \longrightarrow \boxed{ALG_B} \longrightarrow S_B \xrightarrow{g}} \longrightarrow S_A$$

Exemplo

Problema: SELECAO

Entrada: $\langle V, n, k \rangle$, onde $V[1..n]$ é um vetor e $k \in \{1, \ldots, n\}$.
Saída: k-ésimo menor elemento armazenado em V.

Exemplo

Problema: SELECAO

Entrada: $\langle V, n, k \rangle$, onde $V[1..n]$ é um vetor e $k \in \{1, \ldots, n\}$.
Saída: k-ésimo menor elemento armazenado em V.

Problema: ORDENACAO

Entrada: $\langle A, m \rangle$, onde $A[1..m]$ é um vetor.
Saída: vetor B com o mesmo conteúdo de A porém tal que $B[1] \leq B[2] \leq \cdots \leq B[n]$.

SELECAO reduz para ORDENACAO

Precisamos mostrar:

$$\langle V, n, k\rangle \xrightarrow{f} \langle A, m\rangle \longrightarrow ALG_{ORDENACAO} \longrightarrow A \xrightarrow{g} V[\cdot]$$

SELECAO reduz para ORDENACAO

Precisamos mostrar:

$$\langle V, n, k \rangle \xrightarrow{f} \langle A, m \rangle \longrightarrow ALG_{ORDENACAO} \longrightarrow A \xrightarrow{g} V[\cdot]$$

```
1  ALG_selecao(V, n, k) {
2      ALG_ordenacao(V, n);
3      return V[k];
4  }
```

Exemplo

Problema: TAREFAS

Entrada: $\langle T, n, s, f \rangle$, onde $T = \{t_1, \ldots, t_n\}$ é um conjunto de n tarefas onde cada $t_i \in T$ ocorre no intervalo $[s_i, f_i)$.
Saída: conjunto $S \subseteq T$ tal que as tarefas de S são mutuamente compatíveis e $|S|$ é máximo.

Exemplo

Problema: TAREFAS

Entrada: $\langle T, n, s, f \rangle$, onde $T = \{t_1, \ldots, t_n\}$ é um conjunto de n tarefas onde cada $t_i \in T$ ocorre no intervalo $[s_i, f_i)$.
Saída: conjunto $S \subseteq T$ tal que as tarefas de S são mutuamente compatíveis e $|S|$ é máximo.

Problema: CONJINDEPENDENTE

Entrada: $\langle G \rangle$, onde G é um grafo.
Saída: $I \subseteq V(G)$ tal que para todo par $u, v \in I$, $uv \notin E(G)$ e $|I|$ é máximo.

TAREFAS reduz para CONJINDEPENDENTE

Precisamos mostrar:

$$\langle T, n, s, f \rangle \xrightarrow{f} \langle G \rangle \longrightarrow ALG_{CONJINDEPENDENTE} \longrightarrow I \xrightarrow{g} S$$

TAREFAS reduz para CONJINDEPENDENTE

Precisamos mostrar:

$$\langle T, n, s, f \rangle \xrightarrow{f} \langle G \rangle \longrightarrow ALG_{CONJINDEPENDENTE} \longrightarrow I \xrightarrow{g} S$$

```
1  ALG_tarefas(T, n, s, f) {
2      V(G) = T;
3      E(G) = { {ti,tj} : s[i] < f[j] e s[j] < f[i]};
4      S = ALG_conjindependente(G);
5      return S;
6  }
```

Resumo

- Redução é uma forma de criar um algoritmo para o problema A.
  - Por meio de uma redução para B e então usando algoritmos para B.
- Observe que isso não impede que haja outros algoritmos para A.

Complexidade computacional

Problemas de decisão

Otimização

Dadas restrições e uma função objetivo que determina o valor de cada solução, encontrar uma solução melhor valor de função objetivo (maximização ou minimização).

Otimização

Dadas restrições e uma função objetivo que determina o valor de cada solução, encontrar uma solução melhor valor de função objetivo (maximização ou minimização).

Problema: MOCHILAOPT

Entrada: $\langle I, n, v, w, W \rangle$, onde $I = \{1, \ldots, n\}$, v_i é o valor e w_i é o peso do item $i \in I$ e W é a capacidade de peso da mochila.
Saída: subconjunto de itens $S \subseteq I$ com $\sum_{i \in S} w_i \leq W$ e $\sum_{i \in S} v_i$ máximo.

Problemas de decisão

Decisão

Dadas restrições e uma pergunta, responder `sim` ou `não`.

Decisão

Dadas restrições e uma pergunta, responder `sim` ou `não`.

Problema: MOCHILA

Entrada: $\langle I, n, v, w, W, V \rangle$, onde $I = \{1, \ldots, n\}$, v_i é o valor e w_i é o peso do item $i \in I$, W é a capacidade de peso da mochila e V é um número.
Decisão: existe um subconjunto de itens $S \subseteq I$ com $\sum_{i \in S} w_i \leq W$ e $\sum_{i \in S} v_i \geq V$?

Equivalência

- Problemas de otimização e decisão correspondentes são equivalentes no sentido de que se resolvermos um deles, então resolvemos o outro.

Equivalência

- Problemas de otimização e decisão correspondentes são equivalentes no sentido de que se resolvermos um deles, então resolvemos o outro.
    - Podem ser reduzidos entre si!

MOCHILA reduz para MOCHILAOPT

Suponha que sabemos resolver MOCHILAOPT em $\langle I, n, v, w, W \rangle$.
Conseguimos resolver MOCHILA?

MOCHILA reduz para MOCHILAOPT

Suponha que sabemos resolver MOCHILAOPT em $\langle I, n, v, w, W \rangle$.
Conseguimos resolver MOCHILA?

```
1  ALG_mochila(I, n, v, w, W, V) {
2      z = ALG_mochilaopt(I, n, v, w, W);
3      if (z >= V)
4          return "sim";
5      else
6          return "não";
7  }
```

MOCHILAOPT reduz para MOCHILA

Agora suponha que sabemos resolver MOCHILA em $\langle I, n, v, w, W, V \rangle$. Conseguimos resolver MOCHILAOPT?

MOCHILAOPT reduz para MOCHILA

Agora suponha que sabemos resolver MOCHILA em $\langle I, n, v, w, W, V \rangle$. Conseguimos resolver MOCHILAOPT?

```
1  ALG_mochilaopt(I, n, v, w, W) {
2      ini = 0;
3      fim = n * max{v[1], ..., v[n]};
4
5      while (ini < fim) {
6          meio = (ini + fim) / 2;
7          if (ALG_mochila(I, n, v, w, W, meio) == "sim")
8              ini = meio + 1;
9          else
10             fim = meio - 1;
11     }
12
13     if (ALG_mochila(I, n, v, w, W, ini) == "sim")
14         return ini;
15     else
16         return ini - 1;
17 }
```

Outros problemas de decisão

Problema: ALINHAMENTO

Entrada: $\langle X, m, Y, n, \alpha, p \rangle$, onde $X[1..m]$ e $Y = [1..n]$ são sequências, α é função de pontuação e p é um número.
Decisão: existe um alinhamento de X e Y com pontuação $\geq p$?

Outros problemas de decisão

Problema: ARVOREGERADORA

Entrada: $\langle G, w, W \rangle$, onde G é um grafo, $w\colon E(G) \to \mathbb{R}$ e W é um número.
Decisão: existe conjunto $T \subseteq E(G)$ tal que $G[T]$ é árvore geradora e $\sum_{e \in T} w(e) \leq W$?

Outros problemas de decisão

Problema: BARRA

Entrada: $\langle n, p, k \rangle$, onde n é um inteiro tamanho da barra, vender um pedaço de tamanho i dá lucro p_i e k é um número.
Decisão: existe forma de cortar a barra tal que a soma dos lucros de vendas é $\geq k$?

Outros problemas de decisão

Problema: CAMINHOCURTO

Entrada: $\langle D, w, s, t, k \rangle$, onde D é um digrafo, $w\colon E(D) \to \mathbb{R}$, $s, t \in V(D)$ e $k \in \mathbb{R}$.
Decisão: existe st-caminho em D com peso $\leq k$?

Outros problemas de decisão

Problema: CAMINHOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe caminho hamiltoniano em G?

Outros problemas de decisão

Problema: CAMINHOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe caminho hamiltoniano em G?

Problema: CICLOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe ciclo hamiltoniano em G?

Outros problemas de decisão

Problema: CAMINHOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe caminho hamiltoniano em G?

Problema: CICLOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe ciclo hamiltoniano em G?

Problema: TSP

Entrada: $\langle G, w, k \rangle$, onde G é um grafo, $w\colon E(G) \to \mathbb{R}$ e $k \in \mathbb{R}$.
Decisão: existe ciclo hamiltoniano em G com peso $\leq k$?

Redução

Redução formalmente

Redução polinomial

Sejam A e B dois problemas de decisão.
O problema A é redutível para B se existe um algoritmo eficiente f tal que $f(I_A) = I_B$ e

$$I_A \text{ é } \texttt{sim} \text{ se e somente se } I_B \text{ é } \texttt{sim}.$$

CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Seja $\langle G \rangle$ uma entrada de CAMINHOHAMILT.

CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

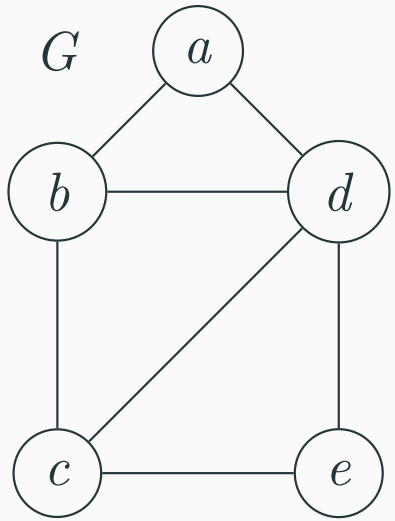
Seja $\langle G \rangle$ uma entrada de CAMINHOHAMILT.

Construa $\langle G' \rangle$ onde $V(G') = V(G) \cup \{x\}$ e $E(G') = E(G) \cup \{xu \colon u \in V(G)\}$. Isso leva tempo $O(|V(G)| + |E(G)|)$, pois basta fazer uma cópia de G e criar um único novo vértice, com aresta para todos os outros.

CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Seja $\langle G \rangle$ uma entrada de CAMINHOHAMILT.

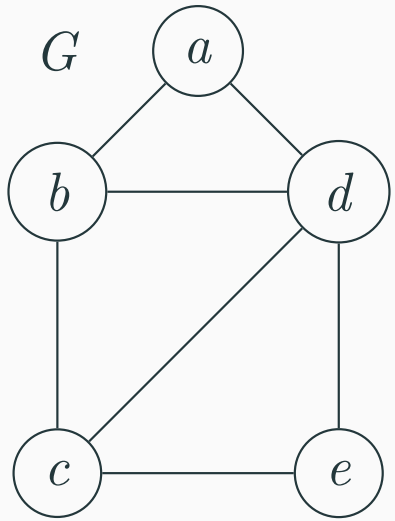
Construa $\langle G' \rangle$ onde $V(G') = V(G) \cup \{x\}$ e $E(G') = E(G) \cup \{xu \colon u \in V(G)\}$. Isso leva tempo $O(|V(G)| + |E(G)|)$, pois basta fazer uma cópia de G e criar um único novo vértice, com aresta para todos os outros.

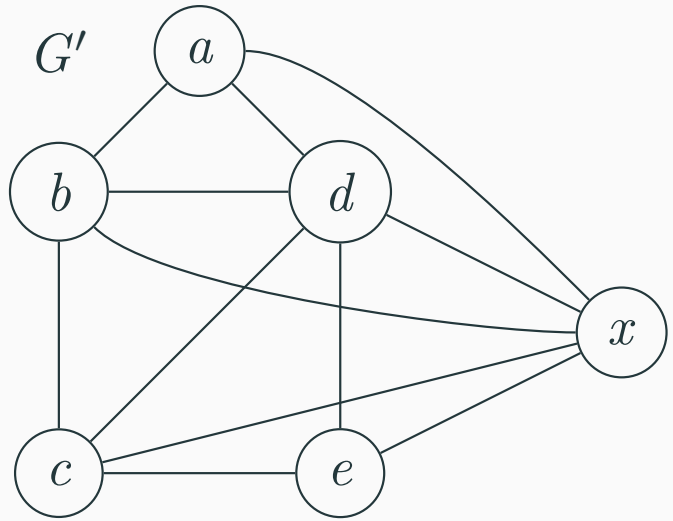


CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Seja $\langle G \rangle$ uma entrada de CAMINHOHAMILT.

Construa $\langle G' \rangle$ onde $V(G') = V(G) \cup \{x\}$ e $E(G') = E(G) \cup \{xu \colon u \in V(G)\}$. Isso leva tempo $O(|V(G)| + |E(G)|)$, pois basta fazer uma cópia de G e criar um único novo vértice, com aresta para todos os outros.

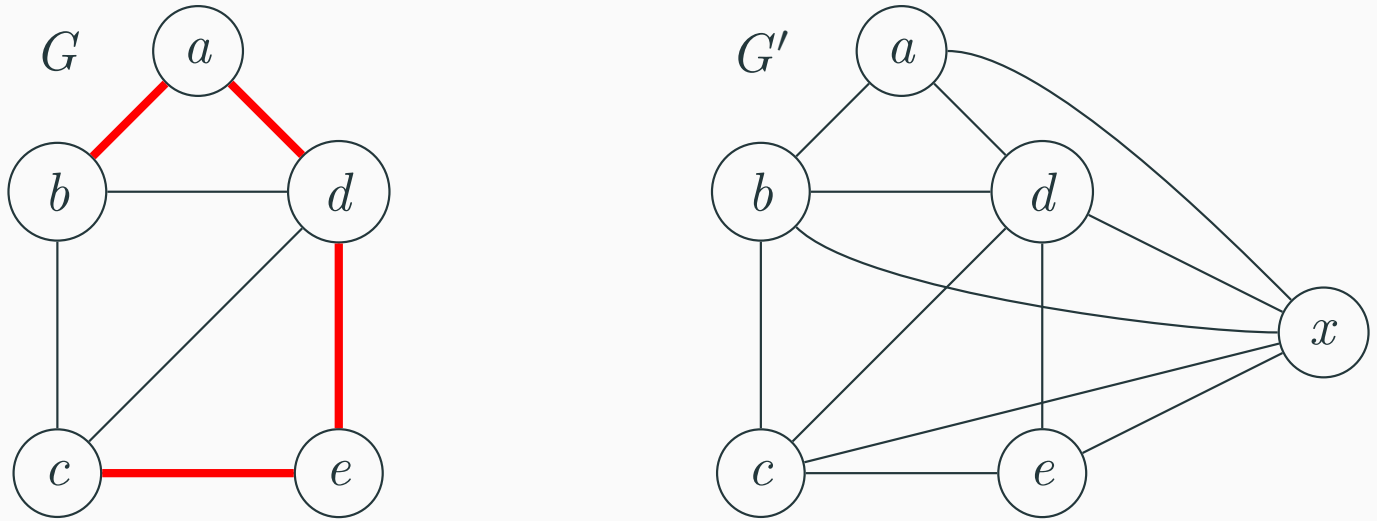




CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Seja $\langle G \rangle$ uma entrada de CAMINHOHAMILT.

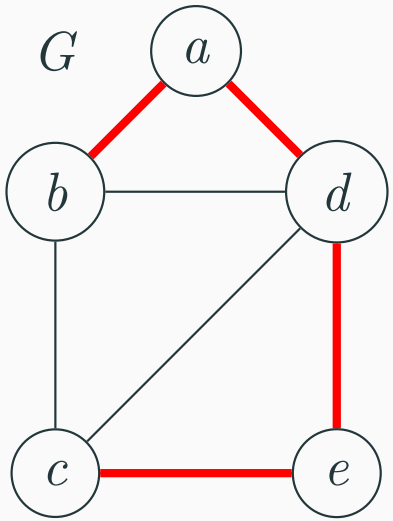
Construa $\langle G' \rangle$ onde $V(G') = V(G) \cup \{x\}$ e $E(G') = E(G) \cup \{xu \colon u \in V(G)\}$. Isso leva tempo $O(|V(G)| + |E(G)|)$, pois basta fazer uma cópia de G e criar um único novo vértice, com aresta para todos os outros.

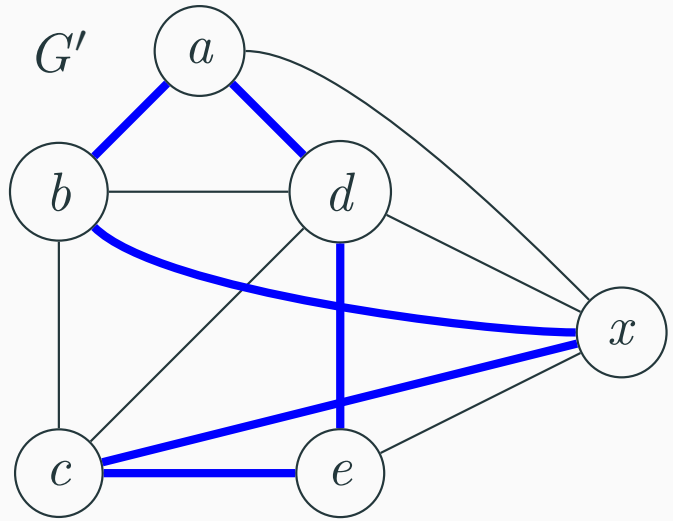


CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Seja $\langle G \rangle$ uma entrada de CAMINHOHAMILT.

Construa $\langle G' \rangle$ onde $V(G') = V(G) \cup \{x\}$ e $E(G') = E(G) \cup \{xu \colon u \in V(G)\}$. Isso leva tempo $O(|V(G)| + |E(G)|)$, pois basta fazer uma cópia de G e criar um único novo vértice, com aresta para todos os outros.





CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Resta mostrar que $\langle G \rangle$ é `sim` para CAMINHOHAMILT se e somente se $\langle G' \rangle$ é `sim` para CICLOHAMILT.

CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Resta mostrar que $\langle G \rangle$ é `sim` para CAMINHOHAMILT se e somente se $\langle G' \rangle$ é `sim` para CICLOHAMILT.

Suponha primeiro que $\langle G \rangle$ é `sim` para CAMINHOHAMILT.
Então existe caminho $C = (v_1, v_2, \ldots, v_n)$ hamiltoniano em G.
Note que $C' = (v_1, v_2, \ldots, v_n, x, v_1)$ é ciclo hamiltoniano em G'.
Então $\langle G' \rangle$ é `sim` para CICLOHAMILT.

CAMINHOHAMILT reduz polinomialmente para CICLOHAMILT

Resta mostrar que $\langle G \rangle$ é `sim` para CAMINHOHAMILT se e somente se $\langle G' \rangle$ é `sim` para CICLOHAMILT.

Suponha primeiro que $\langle G \rangle$ é `sim` para CAMINHOHAMILT.
Então existe caminho $C = (v_1, v_2, \ldots, v_n)$ hamiltoniano em G.
Note que $C' = (v_1, v_2, \ldots, v_n, x, v_1)$ é ciclo hamiltoniano em G'.
Então $\langle G' \rangle$ é `sim` para CICLOHAMILT.

Suponha agora que $\langle G' \rangle$ é `sim` para CICLOHAMILT.
Então existe ciclo $C = (x, v_1, v_2, \ldots, v_n, x)$ hamiltoniano em G'.
Note que $C' = (v_1, v_2, \ldots, v_n)$ é caminho hamiltoniano em G.
Então $\langle G \rangle$ é `sim` para CAMINHOHAMILT.

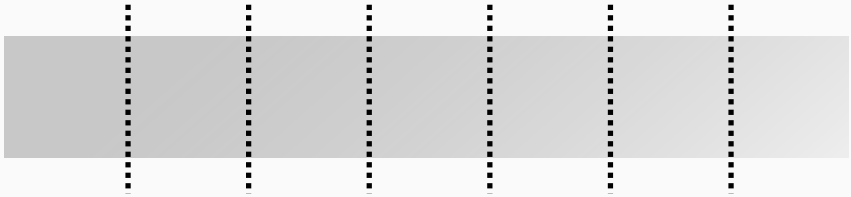
BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

$n = 7$



$p_1 = 3$ $\quad$ $p_2 = 5$ $\quad$ $p_3 = 2$

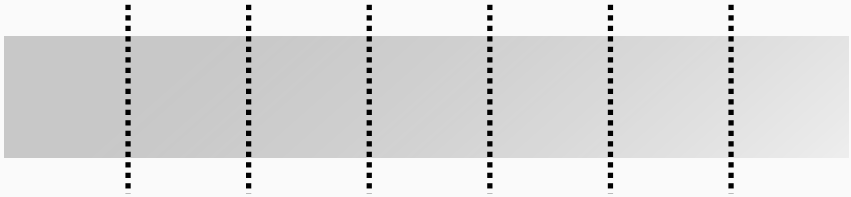
$p_4 = 6$ $\quad$ $p_5 = 1$ $\quad$ $p_6 = 9$

$p_7 = 4$ $\quad$ $k = 50$

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

$n = 7$



$p_1 = 3$ $\quad$ $p_2 = 5$ $\quad$ $p_3 = 2$

$p_4 = 6$ $\quad$ $p_5 = 1$ $\quad$ $p_6 = 9$

$p_7 = 4$ $\quad$ $k = 9$

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

$n = 7$

$p_1 = 3$ $\quad p_2 = 5$ $\quad p_3 = 2$

$p_4 = 6$ $\quad p_5 = 1$ $\quad p_6 = 9$

$p_7 = 4$ $\quad k = 9$

$$(2, 3, 1, 1) = (c_1, c_2, c_3, c_4)$$

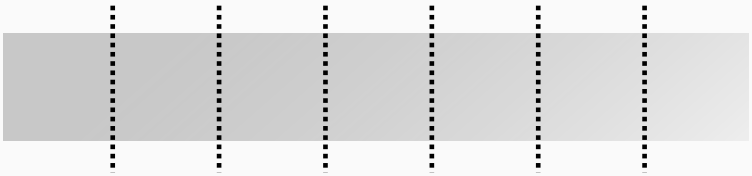
$$\sum_{i=1}^{4} c_i = n = 7$$

$$\sum_{i=1}^{4} p_{c_i} \geq k = 9$$

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

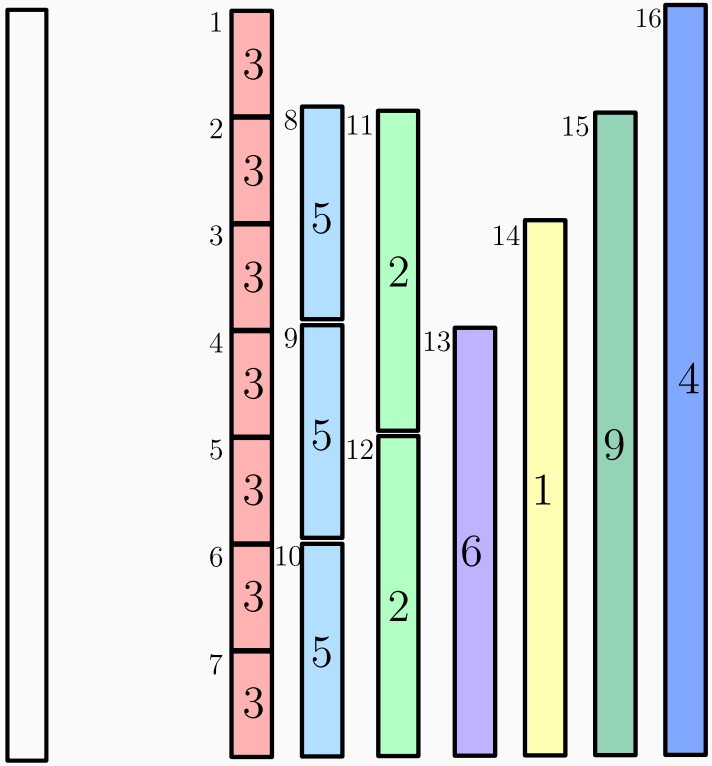
$n = 7$



$p_1 = 3$ $p_2 = 5$ $p_3 = 2$

$p_4 = 6$ $p_5 = 1$ $p_6 = 9$

$p_7 = 4$ $k = 9$



$W = 7$

$V = 9$

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

$n = 7$

$p_1 = 3$ $p_2 = 5$ $p_3 = 2$

$p_4 = 6$ $p_5 = 1$ $p_6 = 9$

$p_7 = 4$ $k = 9$

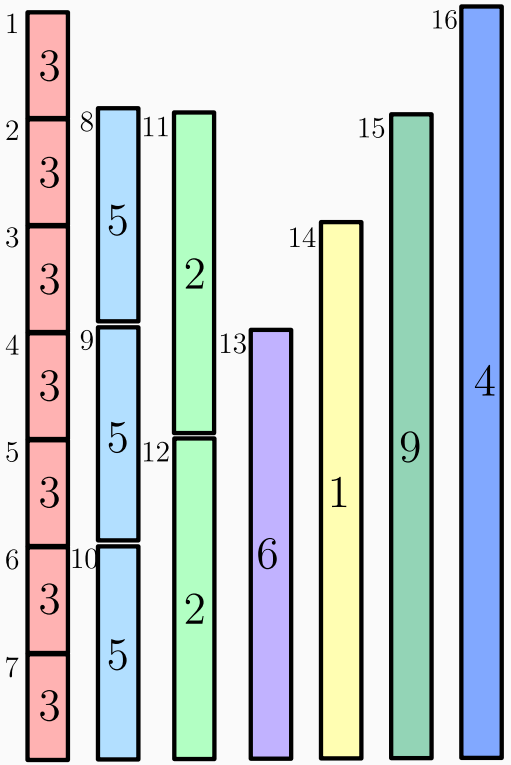
$(2, 3, 1, 1) = (c_1, c_2, c_3, c_4)$

$\sum_{i=1}^{4} c_i = n = 7$

$\sum_{i=1}^{4} p_{c_i} \geq k = 9$

$W = 7$

$V = 9$



$S = \{1, 4, 8, 11\}$

$\sum_{i \in S} w_i \leq W = 7$

$\sum_{i \in S} v_i \geq V = 9$

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

Construa $\langle I, m, v, w, W, V \rangle$ para MOCHILA da seguinte forma:

- crie $\lfloor n/i \rfloor$ itens de peso i e valor p_i cada, correspondentes a um pedaço de tamanho i da barra, $1 \leq i \leq n$;
- $m = \sum_{i=1}^{n} \lfloor n/i \rfloor$ e $I = \{1, \ldots, m\}$;
- $W = n$;
- $V = k$.

Note que isso é feito em tempo $O(n)$.

BARRA reduz polinomialmente para MOCHILA

Seja $\langle n, p, k \rangle$ uma entrada de BARRA.

Construa $\langle I, m, v, w, W, V \rangle$ para MOCHILA da seguinte forma:

- crie $\lfloor n/i \rfloor$ itens de peso i e valor p_i cada, correspondentes a um pedaço de tamanho i da barra, $1 \leq i \leq n$;
- $m = \sum_{i=1}^{n} \lfloor n/i \rfloor$ e $I = \{1, \ldots, m\}$;
- $W = n$;
- $V = k$.

Note que isso é feito em tempo $O(n)$.

Resta mostrar que $\langle n, p, k \rangle$ é `sim` para BARRA se e somente se $\langle I, m, v, w, W, V \rangle$ é `sim` para MOCHILA.

BARRA reduz polinomialmente para MOCHILA

Suponha que $\langle n, p, k \rangle$ é `sim` para BARRA.

Então existem pedaços $(c_1, c_2, \ldots, c_x)$ tais que $\sum_{i=1}^{x} c_i = n$ e $\sum_{i=1}^{x} p_{c_i} \geq k$.
Para cada pedaço c_i, coloque o item j correspondente ao mesmo em um conjunto S.
Note que $\sum_{j \in S} w_j = \sum_{i=1}^{x} c_i = n = W$ e
$\sum_{j \in S} v_j = \sum_{i=1}^{x} p_{c_i} \geq k = V$.

Então $\langle I, m, v, w, W, V \rangle$ é `sim` para MOCHILA.

BARRA reduz polinomialmente para MOCHILA

Suponha agora que $\langle I, m, v, w, W, V \rangle$ é `sim` para MOCHILA.

Então existe conjunto $S \subseteq I$ de itens tais que $\sum_{j \in S} w_j \leq W$ e $\sum_{j \in S} v_j \geq V$.
Para cada item $j \in S$, corte a barra em um tamanho i correspondente ao mesmo.
Sejam $(c_1, c_2, \ldots, c_{|S|})$ os pedaços cortados da barra.
Note que $\sum_{i=1}^{|S|} c_i = \sum_{j \in S} w_j \leq W = n$ e
$\sum_{i=1}^{|S|} p_{c_i} = \sum_{j \in S} v_j \geq V = k$.

Então $\langle n, p, k \rangle$ é `sim` para BARRA.

Outro problema

Problema: SUBSETSUM

Entrada: $\langle A, n, B \rangle$, onde $A = \{s_1, \ldots, s_n\}$ é um conjunto de n inteiros e B é um inteiro.
Decisão: existe $A' \subseteq A$ tal que $\sum_{s \in A'} s = B$?

Outro problema

Problema: SUBSETSUM

Entrada: $\langle A, n, B \rangle$, onde $A = \{s_1, \ldots, s_n\}$ é um conjunto de n inteiros e B é um inteiro.
Decisão: existe $A' \subseteq A$ tal que $\sum_{s \in A'} s = B$?

Exemplos:

$$\langle \{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 2 \rangle$$

$$\langle \{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 20 \rangle$$

$$\langle \{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 37 \rangle$$

$$\langle \{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 74 \rangle$$

SUBSETSUM reduz polinomialmente para MOCHILA

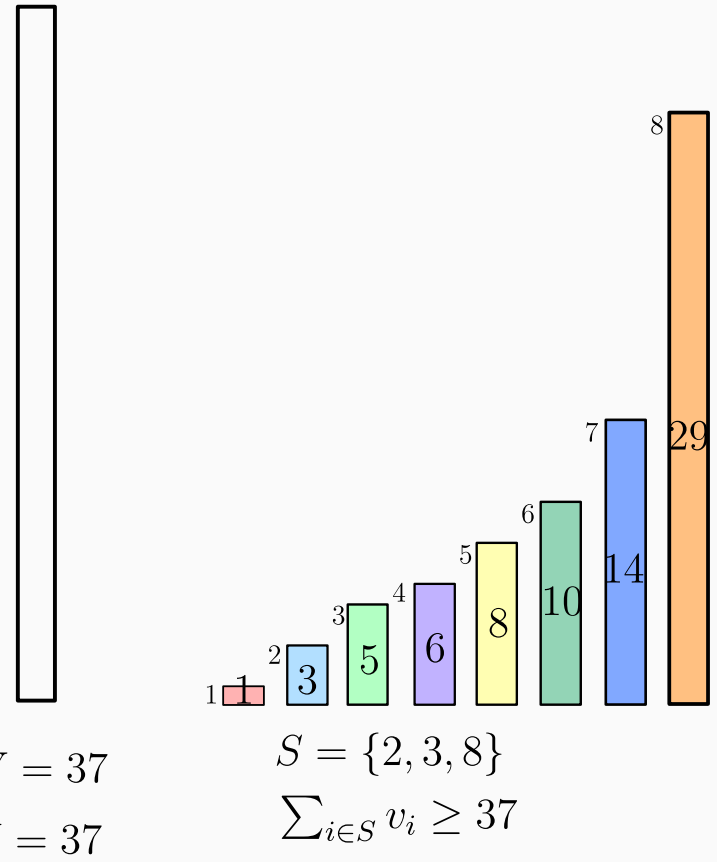
Seja $\langle A, n, B \rangle$ uma entrada de SUBSETSUM.

$$\langle \{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 37 \rangle$$

SUBSETSUM reduz polinomialmente para MOCHILA

Seja $\langle A, n, B\rangle$ uma entrada de SUBSETSUM.

$\langle\{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 37\rangle$



$W = 37$

$V = 37$

$S = \{2, 3, 8\}$

$\sum_{i \in S} v_i \geq 37$

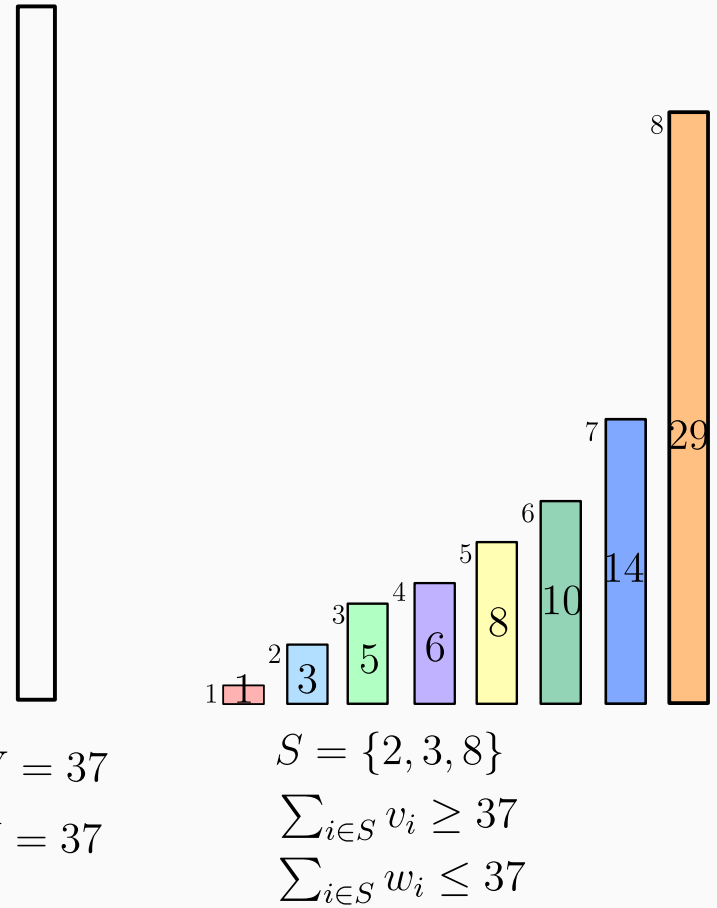
SUBSETSUM reduz polinomialmente para MOCHILA

Seja $\langle A, n, B\rangle$ uma entrada de SUBSETSUM.

$$\langle\{1, 3, 5, 6, 8, 10, 14, 29\}, 8, 37\rangle$$

$$A' = \{3, 5, 29\}$$

$$\sum_{i \in A'} i = 37$$



$W = 37$

$V = 37$

$S = \{2, 3, 8\}$

$\sum_{i \in S} v_i \geq 37$

$\sum_{i \in S} w_i \leq 37$

SUBSETSUM reduz polinomialmente para MOCHILA

Seja $\langle A, n, B \rangle$ uma entrada de SUBSETSUM.

Construa $\langle I, m, v, w, W, V \rangle$ para MOCHILA da seguinte forma:

- crie um item de peso $w = s_i$ e valor $v = s_i$ para cada $s_i \in A$;
- $m = n$ e $I = \{1, \ldots, n\}$;
- $W = B$;
- $V = B$.

Note que isso é feito em tempo $O(n)$.

SUBSETSUM reduz polinomialmente para MOCHILA

Seja $\langle A, n, B\rangle$ uma entrada de SUBSETSUM.

Construa $\langle I, m, v, w, W, V\rangle$ para MOCHILA da seguinte forma:

- crie um item de peso $w = s_i$ e valor $v = s_i$ para cada $s_i \in A$;
- $m = n$ e $I = \{1, \ldots, n\}$;
- $W = B$;
- $V = B$.

Note que isso é feito em tempo $O(n)$.

Resta mostrar que $\langle A, n, B\rangle$ é `sim` para SUBSETSUM se e somente se $\langle I, m, v, w, W, V\rangle$ é `sim` para MOCHILA.

SUBSETSUM reduz polinomialmente para MOCHILA

Suponha que $\langle A, n, B\rangle$ é `sim` para SUBSETSUM.

Então existe $A' \subseteq A$ com $\sum_{s\in A'} s = B$.
Para cada $s \in A'$, coloque o item j correspondente em um conjunto S.
Note que $\sum_{j\in S} w_j = \sum_{s\in A'} s = B = W$ e
$\sum_{j\in S} v_j = \sum_{s\in A'} s = B = V$.

Então $\langle I, m, v, w, W, V\rangle$ é `sim` para MOCHILA.

SUBSETSUM reduz polinomialmente para MOCHILA

Agora suponha que $\langle I, m, v, w, W, V\rangle$ é `sim` para MOCHILA.

Então existe $S \subseteq I$ com $\sum_{j\in S} w_j \leq W$ e $\sum_{j\in S} v_j \geq V$.
Para cada $j \in S$, coloque o valor s_i correspondente em um conjunto A'.
Note que $\sum_{s\in A'} s = \sum_{j\in S} w_j \leq W = B$ e
$\sum_{s\in A'} s = \sum_{j\in S} v_j \geq V = B$.
Mas então só pode ser que $\sum_{s\in A'} s = B$.

Então $\langle A, n, B\rangle$ é `sim` para SUBSETSUM.

Mas qual a importância da redução?

Além de ser uma técnica de projeto de algoritmos, ela é muito utilizada para dar informações sobre a dificuldade de problemas.

Suponha que o problema A reduz em tempo polinomial para o problema B:

$$I_A \longrightarrow \boxed{\xrightarrow{f} I_B \longrightarrow \boxed{ALG_B} \longrightarrow S_B \xrightarrow{g}} \longrightarrow S_A$$

Mas qual a importância da redução?

Além de ser uma técnica de projeto de algoritmos, ela é muito utilizada para dar informações sobre a dificuldade de problemas.

Suponha que o problema A reduz em tempo polinomial para o problema B:

$$I_A \longrightarrow \boxed{\xrightarrow{f} I_B \longrightarrow \boxed{ALG_B} \longrightarrow S_B \xrightarrow{g}} \longrightarrow S_A$$

Que conclusão podemos tirar se:

1. existir algoritmo eficiente para A?
2. existir algoritmo eficiente para B?
3. não existir algoritmo eficiente para A?
4. não existir algoritmo eficiente para B?

Mas qual a importância da redução?

Além de ser uma técnica de projeto de algoritmos, ela é muito utilizada para dar informações sobre a dificuldade de problemas.

Suponha que o problema A reduz em tempo polinomial para o problema B:

$$I_A \longrightarrow \boxed{\xrightarrow{f} I_B \longrightarrow \boxed{ALG_B} \longrightarrow S_B \xrightarrow{g}} \longrightarrow S_A$$

Que conclusão podemos tirar se:

1. existir algoritmo eficiente para A?
2. existir algoritmo eficiente para B?
3. não existir algoritmo eficiente para A?
4. não existir algoritmo eficiente para B?

Ou seja, B é tão difícil quanto A.
Ou então, A não é mais difícil do que B.

Mas qual a importância da redução?

Com exemplos:

- BARRA reduz polinomialmente para MOCHILA.
- SUBSETSUM reduz polinomialmente para MOCHILA.

Sabemos que:

- existe algoritmo eficiente para BARRA;
- (ainda) não existe algoritmo eficiente para SUBSETSUM.

Qual a conclusão sobre MOCHILA?

Classes de complexidade

Classe P

Conjunto de problemas de decisão que podem ser resolvidos por um algoritmo eficiente.

BARRA, ALINHAMENTO, ARVOREGERADORA, CAMINHOCURTO são problemas em P.

Nem todos os problemas do mundo estão em P

Não se sabe!

Até agora, não encontramos algoritmos de tempo polinomial que resolvem MOCHILA, TSP, CICLOHAMILT, CAMINHOHAMILT, SUBSETSUM, ...

Mas eles são facilmente verificáveis

Para convencer alguém que uma instância é `sim`, basta apresentar um certificado positivo.

- $\langle \{2, 4, 5, 7, 9, 10\}, 5, 9 \rangle$ é `sim` para SUBSETSUM?
- $\langle \{2, 4, 5, 7, 9, 10\}, 5, 25 \rangle$ é `sim` para SUBSETSUM?
- $\langle \{1, 2, 3, 4, 5\}, 5, v = (3, 4, 6, 21, 1), w = (3, 4, 6, 2, 7), 10, 15 \rangle$ é `sim` para MOCHILA?
- $\langle \{1, 2, 3, 4, 5\}, 5, v = (3, 4, 6, 21, 1), w = (3, 4, 6, 2, 7), 10, 48 \rangle$ é `sim` para MOCHILA?

Algoritmo verificador

Seja T um problema qualquer.

Um algoritmo A é verificador se:

- para toda I_T que é `sim`, existe um conjunto de dados D tal que $A(I_T, D)$ devolve `sim` em tempo polinomial;
- para toda I_T que é `não`, qualquer conjunto de dados D faz $A(I_T, D)$ devolver `não` em tempo polinomial.

O conjunto D acima é um certificado positivo.

Classe NP

Classe NP

Conjunto de problemas de decisão para os quais existe um algoritmo verificador que aceita um certificado positivo em tempo polinomial.

É o caso de MOCHILA, TSP, CICLOHAMILT, CAMINHOHAMILT, SUBSETSUM, STEINER, ...

Note que $\mathsf{P} \subseteq \mathsf{NP}$.

P vs. NP

Note que $\mathsf{P} \subseteq \mathsf{NP}$.

Será que $\mathsf{NP} \subseteq \mathsf{P}$?

P vs. NP

Note que $\mathsf{P} \subseteq \mathsf{NP}$.

Será que $\mathsf{NP} \subseteq \mathsf{P}$?

P é igual a NP?

P vs. NP

Note que $\mathsf{P} \subseteq \mathsf{NP}$.

Será que $\mathsf{NP} \subseteq \mathsf{P}$?

P é igual a NP?

Um dos problemas do milênio: prêmio de 1 milhão de dólares para quem resolver.

Problemas NP-completos

Discussão

- O que significa dizer que um problema A está em NP?

Problemas NP-completos

Problema NP-completo

Um problema Q é NP-completo se $Q \in \mathsf{NP}$ e todo outro problema de NP se reduz polinomialmente a Q.

O que isso significa?

Problema NP-completo

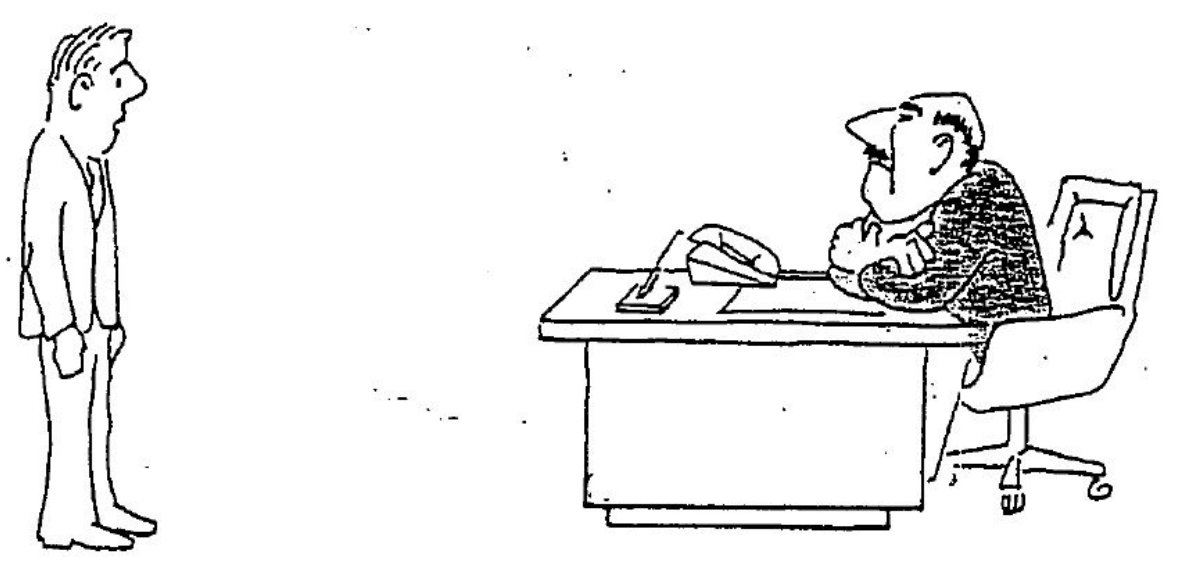
Um problema Q é NP-completo se $Q \in \mathsf{NP}$ e todo outro problema de NP se reduz polinomialmente a Q.

O que isso significa?

Se X é NP-completo, podemos dizer que $\mathsf{P} = \mathsf{NP}$ se e somente se X pertence a P.

Resolva esse problema

Ele está em NP!



"I can't find an efficient algorithm, I guess I'm just too dumb."

Ele é NP-completo!

"I can't find an efficient algorithm, but neither can all these famous people."

Problema NP-difícil

Um problema Q é NP-difícil se todo outro problema de NP se reduz polinomialmente a Q.

Problemas NP-difíceis

Problema NP-difícil

Um problema Q é NP-difícil se todo outro problema de NP se reduz polinomialmente a Q.

Problema NP-completo

Um problema Q é NP-completo se $Q \in \mathsf{NP}$ e Q é NP-difícil.

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Primeiro mostramos que o problema está em NP.

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Primeiro mostramos que o problema está em NP.

Depois enumeramos todos os problemas de NP e fazemos uma redução polinomial de cada um deles para o nosso problema (mostramos que ele é NP-difícil).

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Primeiro mostramos que o problema está em NP.

Depois enumeramos todos os problemas de NP e fazemos uma redução polinomial de cada um deles para o nosso problema (mostramos que ele é NP-difícil).

IMPOSSÍVEL!

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Primeiro mostramos que o problema está em NP.

Depois enumeramos todos os problemas de NP e fazemos uma redução polinomial de cada um deles para o nosso problema (mostramos que ele é NP-difícil).

IMPOSSÍVEL!

Mas a operação de redução pode ser composta!

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Primeiro mostramos que o problema está em NP.

Problemas NP-completos

E como mostrar que um problema é NP-completo?

Primeiro mostramos que o problema está em NP.

Depois encontramos um problema que já é NP-completo e o reduzimos polinomialmente para o nosso.

Problemas NP-completos

Por exemplo:

- Vimos que MOCHILA está em NP.
- Vimos que SUBSETSUM reduz polinomialmente para MOCHILA.
- Sabemos (??) que SUBSETSUM É NP-completo.
- Portanto, MOCHILA é NP-completo!

Problemas NP-completos

Mas e o primeiro problema NP-completo?

Mas e o primeiro problema NP-completo?

Problema: SAT

Entrada: $\langle \phi \rangle$, onde ϕ é uma fórmula Booleana sobre os as variáveis $x_1, \ldots, x_n$.
Decisão: Existe uma atribuição de valores a $x_1, \ldots, x_n$ tal que ϕ é satisfatível?

Mas e o primeiro problema NP-completo?

Problema: SAT

Entrada: $\langle \phi \rangle$, onde ϕ é uma fórmula Booleana sobre os as variáveis $x_1, \ldots, x_n$.
Decisão: Existe uma atribuição de valores a $x_1, \ldots, x_n$ tal que ϕ é satisfatível?

Teorema Cook-Levin: SAT é NP-completo.

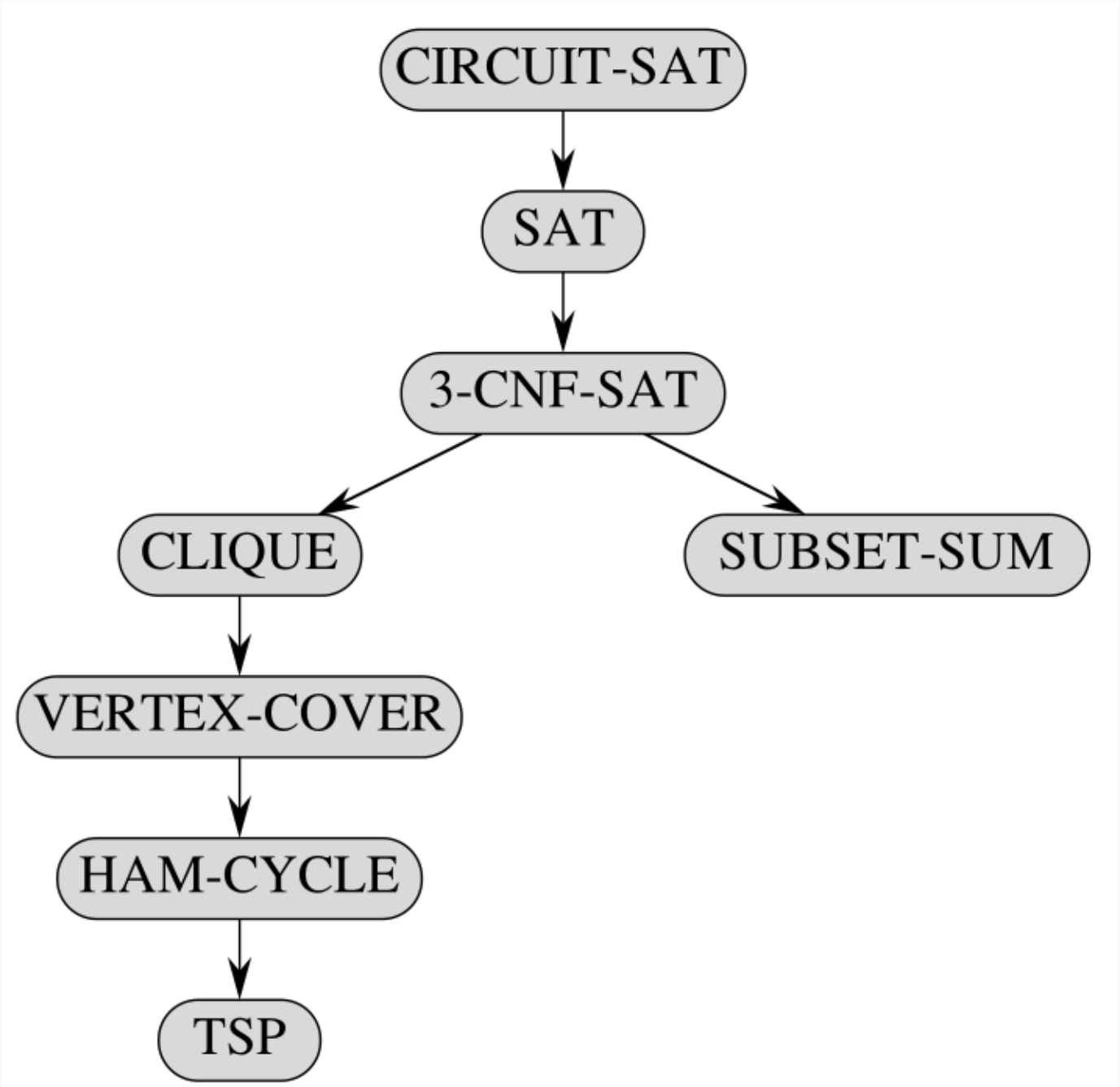
Uma versão mais fácil de lidar.

Problema: 3-SAT

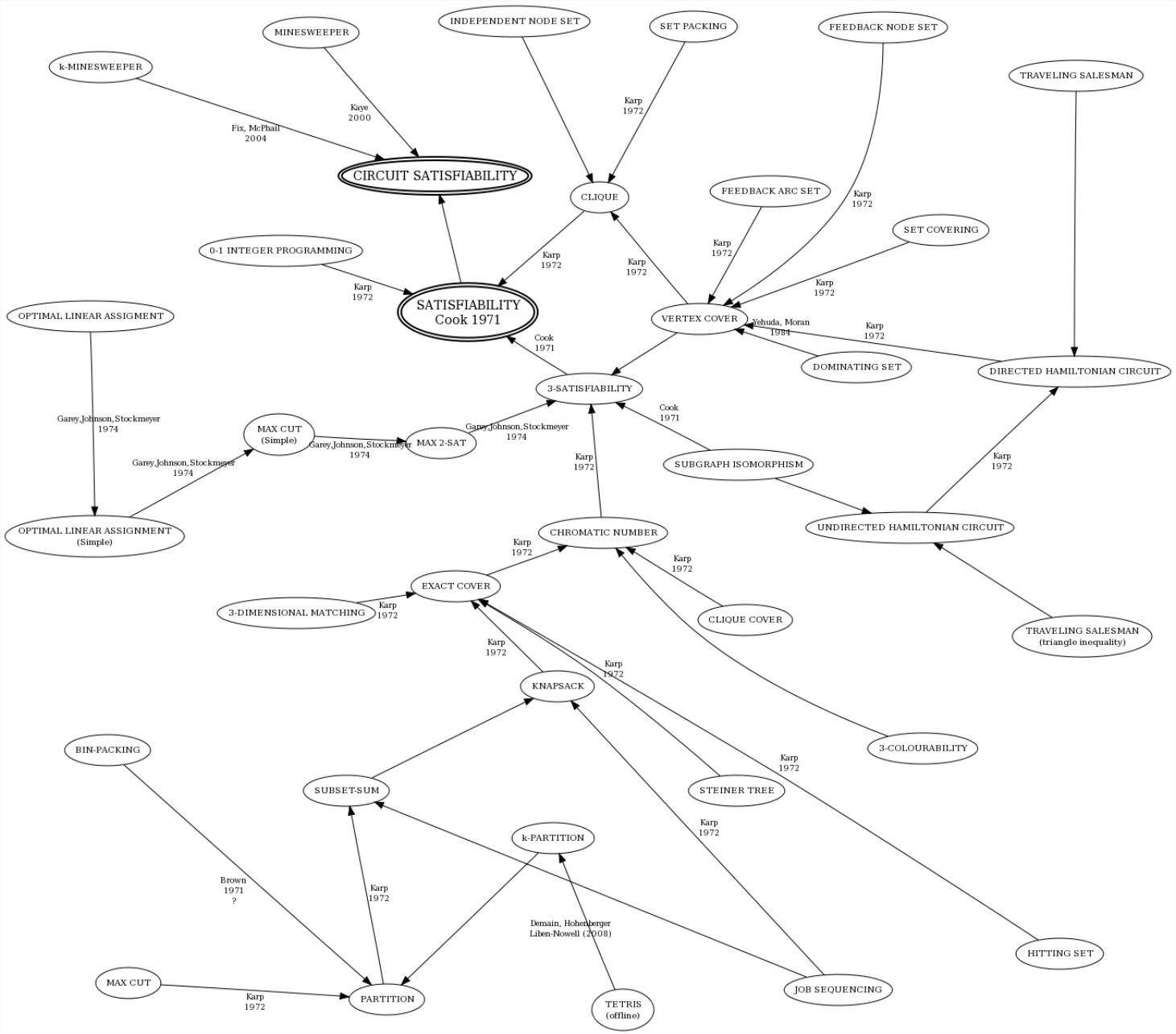
Entrada: $\langle \phi \rangle$, onde ϕ é uma fórmula 3-CNF contendo literais de variáveis booleanas $x_1, \ldots, x_n$.
Decisão: Existe uma atribuição de valores a $x_1, \ldots, x_n$ tal que ϕ é satisfatível?

Por meio de uma redução de SAT para 3-SAT, prova-se que 3-SAT também é NP-completo.

Problemas NP-completos



Problemas NP-completos



Outro problema NP-completo: CLIQUE

Problema: CLIQUE

Entrada: $\langle G, k \rangle$, onde G é um grafo e k é um inteiro positivo.
Decisão: Existe conjunto $S \subseteq V(G)$ tal que para todo par $u, v \in S$ existe aresta $uv \in E(G)$ e $|S| \geq k$?

Outro problema NP-completo: CLIQUE

Problema: CLIQUE

Entrada: $\langle G, k \rangle$, onde G é um grafo e k é um inteiro positivo.
Decisão: Existe conjunto $S \subseteq V(G)$ tal que para todo par $u, v \in S$ existe aresta $uv \in E(G)$ e $|S| \geq k$?

Teorema

CLIQUE é NP-completo.

Demonstração

Primeiro mostramos que CLIQUE está em NP e depois faremos uma redução polinomial de 3-SAT para CLIQUE.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

CLIQUE está em NP pois, dados $\langle G, k \rangle$ e um conjunto S qualquer de vértices, podemos verificar se $|S| \geq k$ e se cada vértice de S, que são $O(|V(G)|)$, é vizinho de todos os outros vértices em S, o que pode ser feito olhando sua vizinhança e portanto leva tempo $O(|V(G)| + |E(G)|)$.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora considere uma fórmula 3-CNF ϕ com m cláusulas sobre literais das variáveis $v_1, \ldots, v_n$.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora considere uma fórmula 3-CNF ϕ com m cláusulas sobre literais das variáveis $v_1, \ldots, v_n$.

Vamos construir um grafo G com $3m$ vértices: para cada cláusula, temos 3 vértices representando seus literais.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora considere uma fórmula 3-CNF ϕ com m cláusulas sobre literais das variáveis $v_1, \ldots, v_n$.

Vamos construir um grafo G com $3m$ vértices: para cada cláusula, temos 3 vértices representando seus literais.

Haverá aresta entre dois vértices x e y se e somente se os literais representados por eles estiverem em cláusulas diferentes, x corresponde a um literal v e y não corresponde ao literal $\bar{v}$.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora considere uma fórmula 3-CNF ϕ com m cláusulas sobre literais das variáveis $v_1, \ldots, v_n$.

Vamos construir um grafo G com $3m$ vértices: para cada cláusula, temos 3 vértices representando seus literais.

Haverá aresta entre dois vértices x e y se e somente se os literais representados por eles estiverem em cláusulas diferentes, x corresponde a um literal v e y não corresponde ao literal $\bar{v}$.

Por fim, tome $k = m$.

Outro problema NP-completo: CLIQUE

Teorema

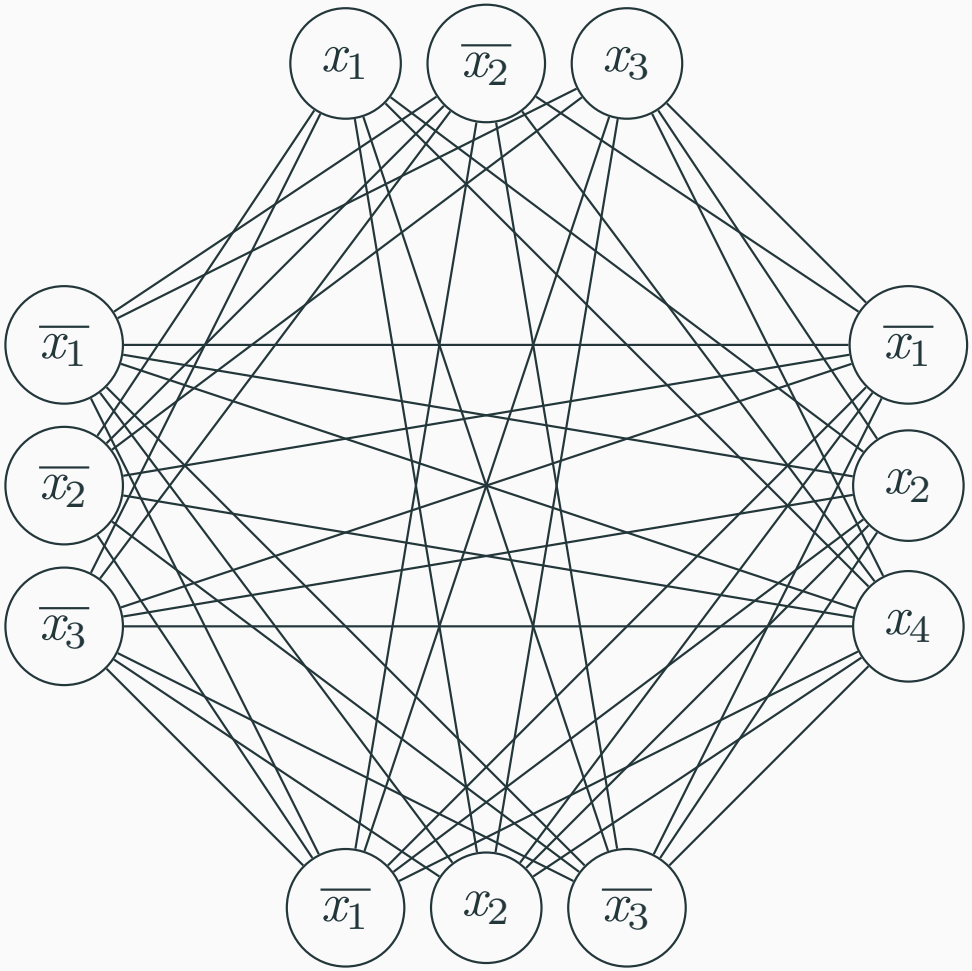
CLIQUE é NP-completo.

Demonstração (continuação)

Agora considere uma fórmula 3-CNF ϕ com m cláusulas sobre literais das variáveis $v_1, \ldots, v_n$.
Vamos construir um grafo G com $3m$ vértices: para cada cláusula, temos 3 vértices representando seus literais.
Haverá aresta entre dois vértices x e y se e somente se os literais representados por eles estiverem em cláusulas diferentes, x corresponde a um literal v e y não corresponde ao literal $\bar{v}$.
Por fim, tome $k = m$.
Temos então uma instância $\langle G, k \rangle$ para CLIQUE gerada em tempo polinomial, pois o grafo tem $3m$ vértices e no máximo $9m^2$ arestas.
Resta verificar se $\langle \phi \rangle$ é `sim` para 3-SAT se e somente se $\langle G, k \rangle$ é `sim` par CLIQUE.

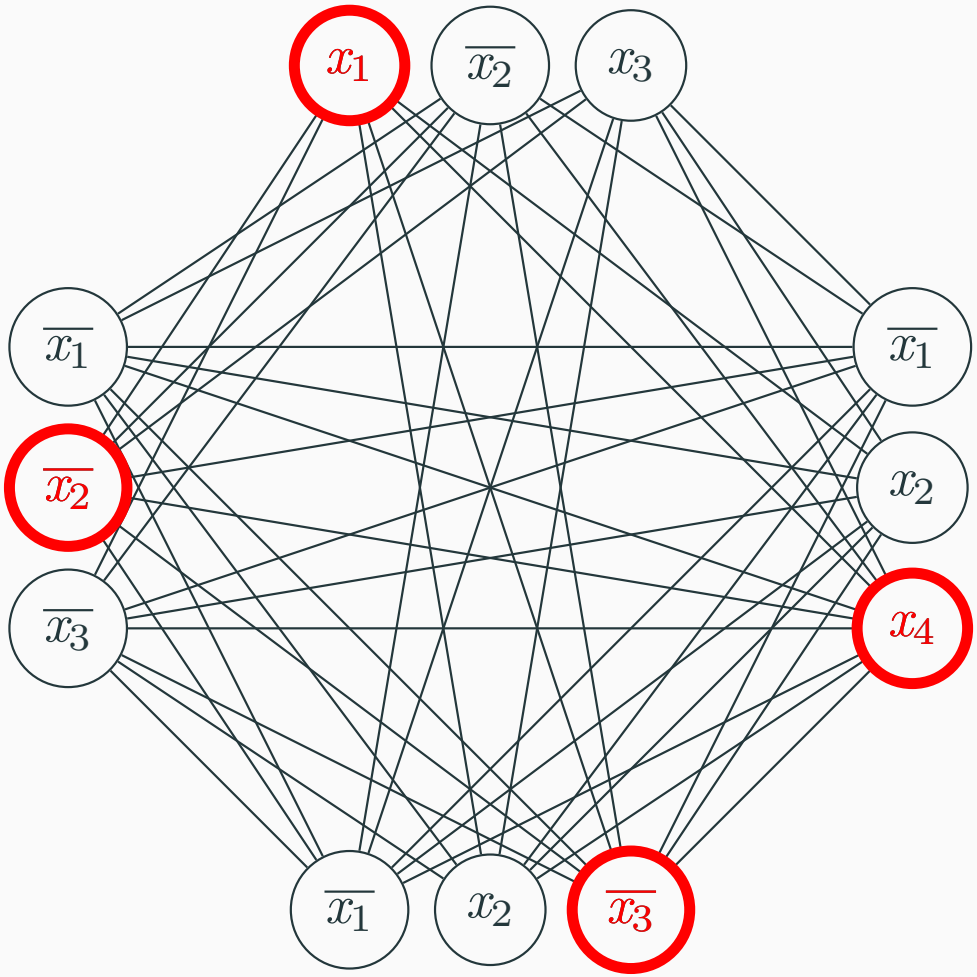
Outro problema NP-completo: CLIQUE

Dada $\phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$, construa:



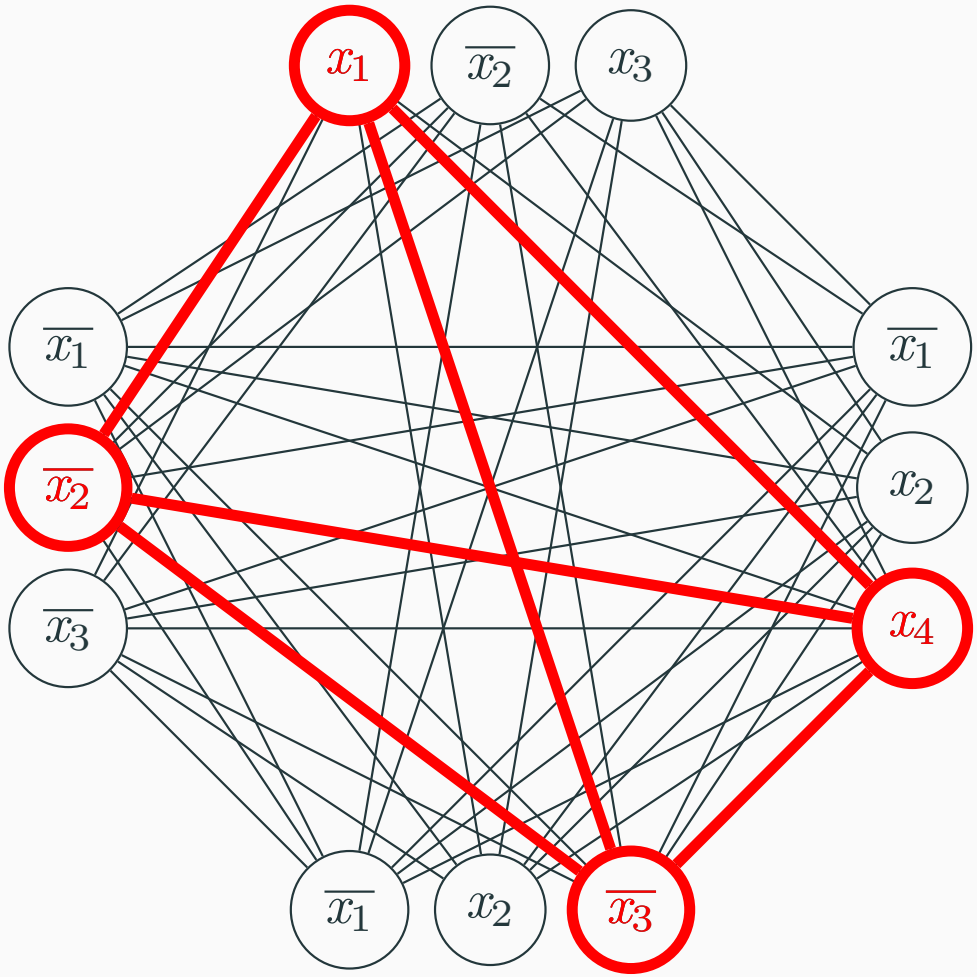
Outro problema NP-completo: CLIQUE

Dada $\phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$, construa:



Outro problema NP-completo: CLIQUE

Dada $\phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$, construa:



Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Suponha que $\langle \phi \rangle$ é `sim` para 3-SAT. Então ϕ é satisfatível.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Suponha que $\langle \phi \rangle$ é `sim` para 3-SAT. Então ϕ é satisfatível.
Então em cada cláusula, pelo menos um literal tem valor `true`. Tome os vértices correspondentes a esses literais em um conjunto S.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Suponha que $\langle \phi \rangle$ é `sim` para 3-SAT. Então ϕ é satisfatível.
Então em cada cláusula, pelo menos um literal tem valor `true`. Tome os vértices correspondentes a esses literais em um conjunto S.
Como um literal e sua negação não podem ter valor `true`, existe uma aresta entre quaisquer dois vértices que representam esses literais em G.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Suponha que $\langle \phi \rangle$ é `sim` para 3-SAT. Então ϕ é satisfatível.
Então em cada cláusula, pelo menos um literal tem valor `true`. Tome os vértices correspondentes a esses literais em um conjunto S.
Como um literal e sua negação não podem ter valor `true`, existe uma aresta entre quaisquer dois vértices que representam esses literais em G.
Então S é uma clique em G com pelo menos m vértices, já que pegamos pelo menos um por cláusula, e note que $m = k$.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Suponha que $\langle \phi \rangle$ é `sim` para 3-SAT. Então ϕ é satisfatível.
Então em cada cláusula, pelo menos um literal tem valor `true`. Tome os vértices correspondentes a esses literais em um conjunto S.
Como um literal e sua negação não podem ter valor `true`, existe uma aresta entre quaisquer dois vértices que representam esses literais em G.
Então S é uma clique em G com pelo menos m vértices, já que pegamos pelo menos um por cláusula, e note que $m = k$. Logo, $\langle G, k \rangle$ é `sim` para CLIQUE.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então existe conjunto $S \subseteq V(G)$ que é clique e $|S| \geq k$.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então existe conjunto $S \subseteq V(G)$ que é clique e $|S| \geq k$.
Então existe uma aresta entre quaisquer dois vértices de S.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então existe conjunto $S \subseteq V(G)$ que é clique e $|S| \geq k$.
Então existe uma aresta entre quaisquer dois vértices de S.
Então esses dois não representam literais que são a negação um do outro e eles estão em cláusulas diferentes.

Outro problema NP-completo: CLIQUE

Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então existe conjunto $S \subseteq V(G)$ que é clique e $|S| \geq k$.
Então existe uma aresta entre quaisquer dois vértices de S.
Então esses dois não representam literais que são a negação um do outro e eles estão em cláusulas diferentes.
Então dar valor `true` aos literais representados pelos vértices de S satisfaz ϕ.

Outro problema NP-completo: CLIQUE

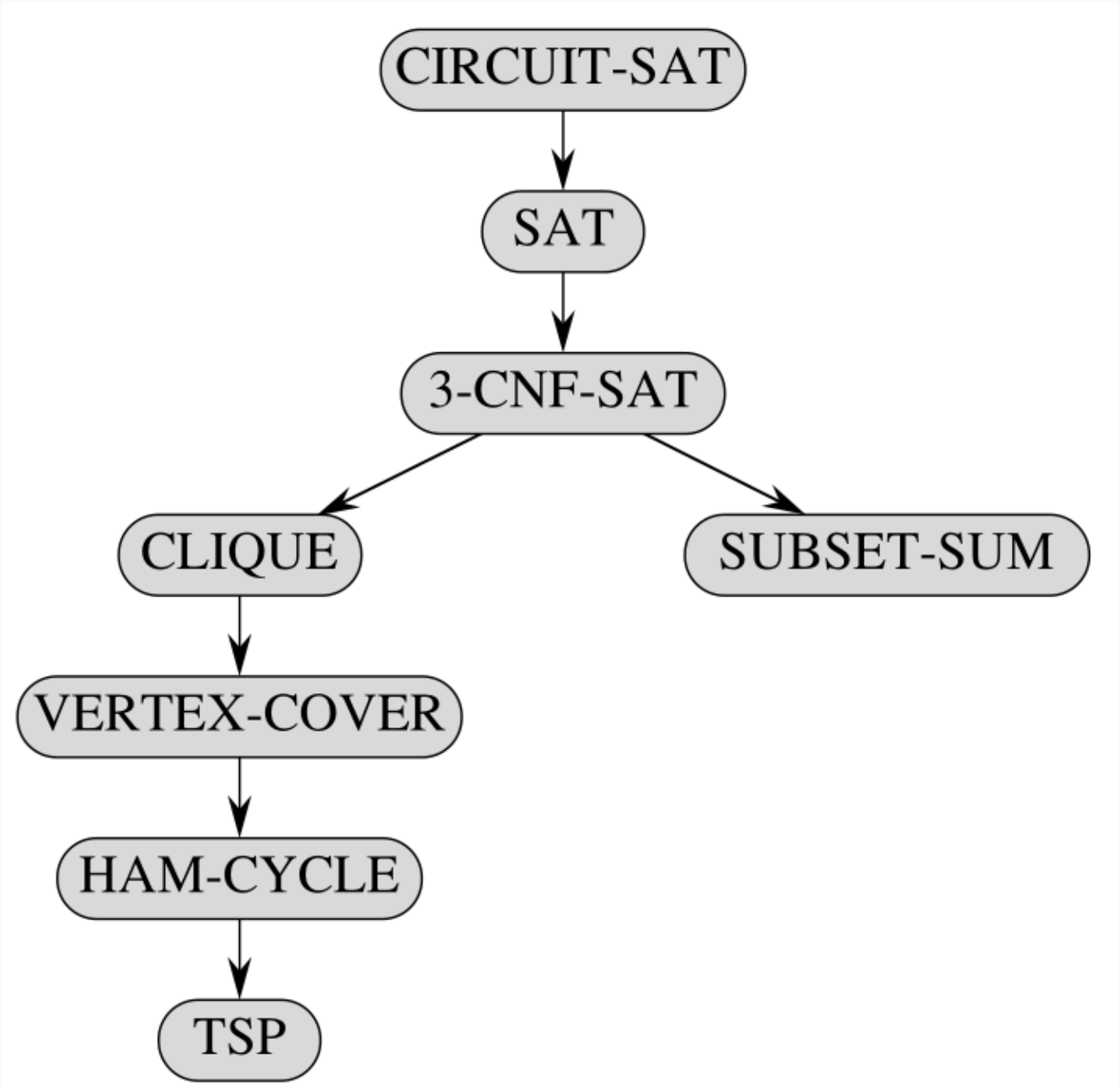
Teorema

CLIQUE é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então existe conjunto $S \subseteq V(G)$ que é clique e $|S| \geq k$.
Então existe uma aresta entre quaisquer dois vértices de S.
Então esses dois não representam literais que são a negação um do outro e eles estão em cláusulas diferentes.
Então dar valor `true` aos literais representados pelos vértices de S satisfaz ϕ. Logo, $\langle \phi \rangle$ é `sim` para 3-SAT. C.Q.D.

Classe NP-completo



Outro problema NP-completo: VERTEXCOVER

Uma cobertura por vértices é um conjunto $S \subseteq V(G)$ tal que toda aresta $uv \in E(G)$ tem $u \in S$ ou $v \in S$.

Problema: VERTEXCOVER

Entrada: $\langle G, k \rangle$, onde G é um grafo e k é um inteiro.
Decisão: Existe uma cobertura por vértices $S \subseteq V(G)$ tal que $|S| \leq k$?

Outro problema NP-completo: VERTEXCOVER

Uma cobertura por vértices é um conjunto $S \subseteq V(G)$ tal que toda aresta $uv \in E(G)$ tem $u \in S$ ou $v \in S$.

Problema: VERTEXCOVER

Entrada: $\langle G, k \rangle$, onde G é um grafo e k é um inteiro.
Decisão: Existe uma cobertura por vértices $S \subseteq V(G)$ tal que $|S| \leq k$?

Teorema

VERTEXCOVER é NP-completo.

Demonstração

Primeiro mostramos que VERTEXCOVER está em NPe depois faremos uma redução polinomial de CLIQUE para VERTEXCOVER.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

VERTEXCOVER está em NP pois, dado um conjunto S de vértices, pode-se verificar se $|S| \leq k$ e se todas as arestas de G têm ao menos um extremo em S em tempo polinomial, bastando percorrer S, cujo tamanho é $O(|V(G)|)$, e cada aresta de G.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora considere um grafo G e um valor k inteiro, instância para CLIQUE.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora considere um grafo G e um valor k inteiro, instância para CLIQUE.

O grafo complemento de G é o grafo $\overline{G}$, em que $V(\overline{G}) = V(G)$ e $E(\overline{G}) = \{uv \colon uv \notin E(G)\}$.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora considere um grafo G e um valor k inteiro, instância para CLIQUE.

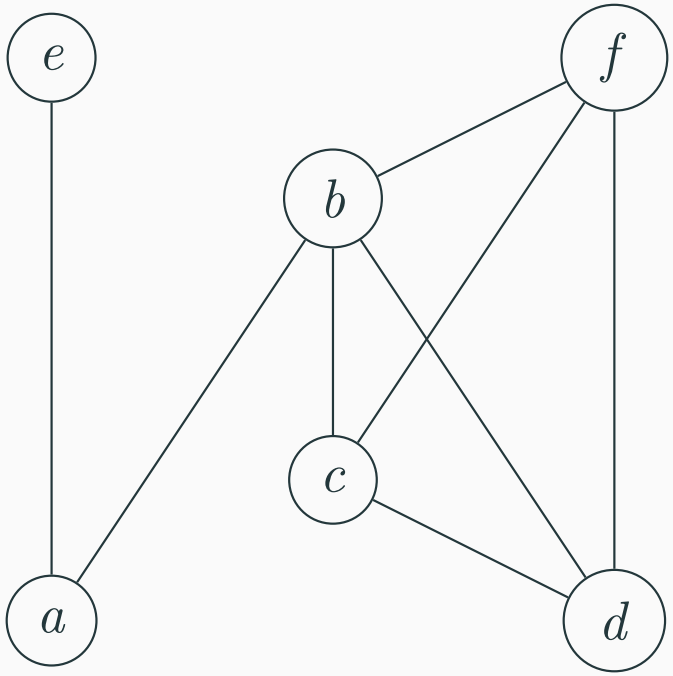
O grafo complemento de G é o grafo $\overline{G}$, em que $V(\overline{G}) = V(G)$ e $E(\overline{G}) = \{uv : uv \notin E(G)\}$.

Tomando $t = |V(G)| - k$, temos então uma instância $\langle \overline{G}, t \rangle$ para VERTEXCOVER construída em tempo tempo polinomial no tamanho de G.

Resta verificar que $\langle G, k \rangle$ é `sim` para CLIQUE se e somente se $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER.

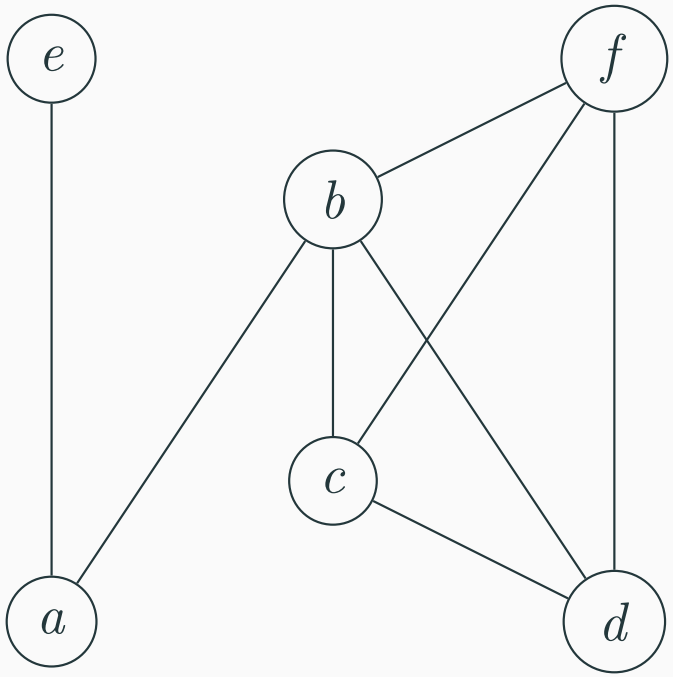
Outro problema NP-completo: VERTEXCOVER

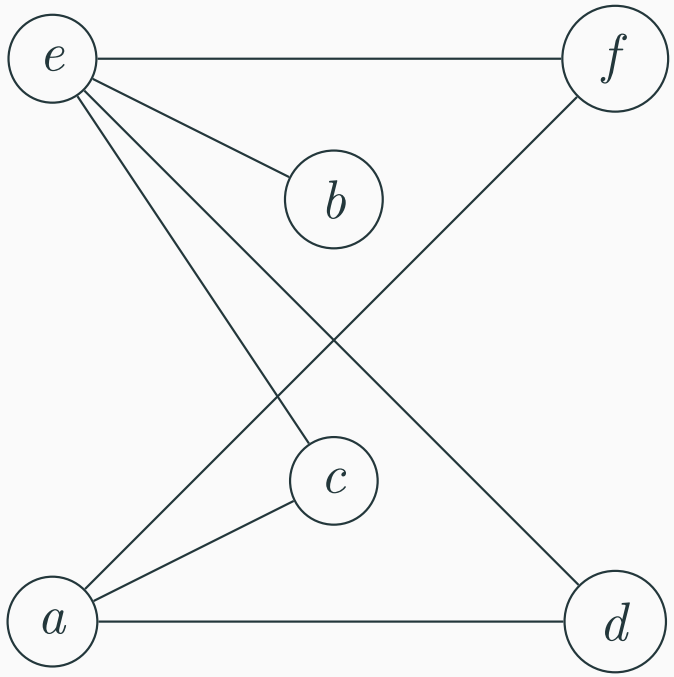
Dado G, considere $\overline{G}$:



Outro problema NP-completo: VERTEXCOVER

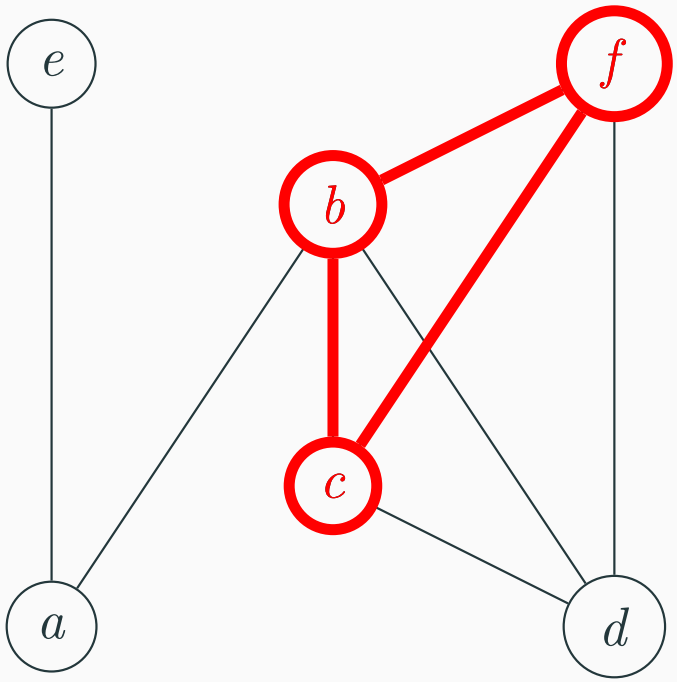
Dado G, considere $\overline{G}$:

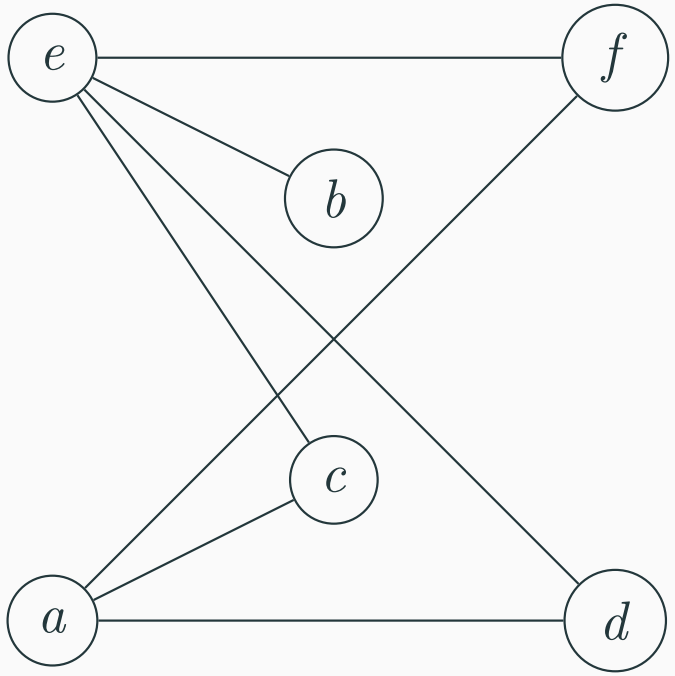




Outro problema NP-completo: VERTEXCOVER

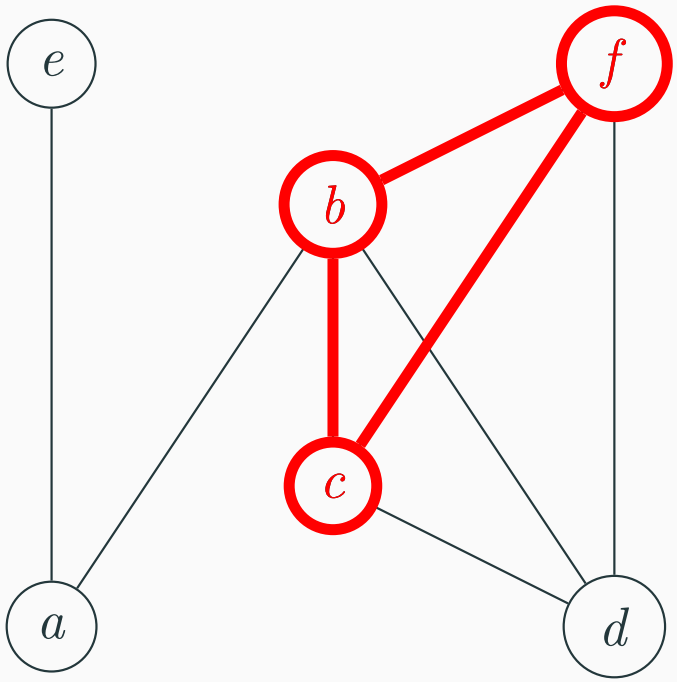
Dado G, considere $\overline{G}$:

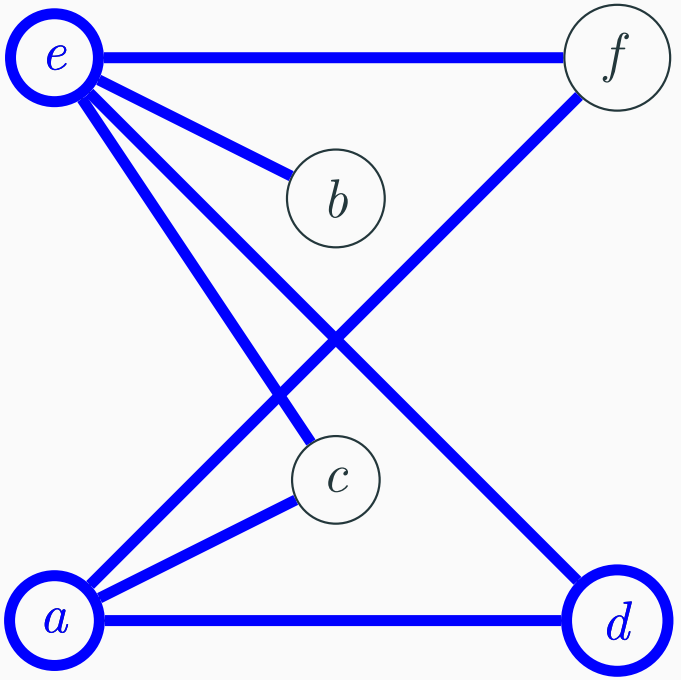




Outro problema NP-completo: VERTEXCOVER

Dado G, considere $\overline{G}$:





Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então G contém uma clique S de tamanho pelo menos k.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então G contém uma clique S de tamanho pelo menos k.
Isso significa que para todo par $u, v \in S$ temos $uv \in E(G)$, o que implica em $uv \notin E(\overline{G})$.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Suponha que $\langle G, k \rangle$ é `sim` para CLIQUE. Então G contém uma clique S de tamanho pelo menos k.
Isso significa que para todo par $u, v \in S$ temos $uv \in E(G)$, o que implica em $uv \notin E(\overline{G})$.
Então para toda aresta $xy \in E(\overline{G})$, devemos ter que $x \notin S$ ou $y \notin S$, isto é, $x \in V(G) \setminus S$ ou $y \in V(G) \setminus S$.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Suponha que $\langle G, k\rangle$ é `sim` para CLIQUE. Então G contém uma clique S de tamanho pelo menos k.
Isso significa que para todo par $u, v \in S$ temos $uv \in E(G)$, o que implica em $uv \notin E(\overline{G})$.
Então para toda aresta $xy \in E(\overline{G})$, devemos ter que $x \notin S$ ou $y \notin S$, isto é, $x \in V(G) \setminus S$ ou $y \in V(G) \setminus S$.
Logo, $V(G) \setminus S$ é uma cobertura por vértices de $\overline{G}$.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Suponha que $\langle G, k\rangle$ é `sim` para CLIQUE. Então G contém uma clique S de tamanho pelo menos k.
Isso significa que para todo par $u, v \in S$ temos $uv \in E(G)$, o que implica em $uv \notin E(\overline{G})$.
Então para toda aresta $xy \in E(\overline{G})$, devemos ter que $x \notin S$ ou $y \notin S$, isto é, $x \in V(G) \setminus S$ ou $y \in V(G) \setminus S$.
Logo, $V(G) \setminus S$ é uma cobertura por vértices de $\overline{G}$.
Como $|S| \geq k$, temos $|V(G) \setminus S| = |V(G)| - |S| \leq |V(G)| - k = t$.
Então $\langle \overline{G}, t\rangle$ é `sim` para VERTEXCOVER.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER. Então $\overline{G}$ contém uma cobertura por vértices S de tamanho no máximo t.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER. Então $\overline{G}$ contém uma cobertura por vértices S de tamanho no máximo t.
Isso significa que para toda aresta $uv \in E(\overline{G})$, temos $u \in S$ ou $v \in S$.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER. Então $\overline{G}$ contém uma cobertura por vértices S de tamanho no máximo t.
Isso significa que para toda aresta $uv \in E(\overline{G})$, temos $u \in S$ ou $v \in S$.
A contrapositiva disso nos diz que se $u \notin S$ e $v \notin S$, então $uv \notin E(\overline{G})$, o que por sua vez implica em $uv \in E(G)$.

Outro problema NP-completo: VERTEXCOVER

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER. Então $\overline{G}$ contém uma cobertura por vértices S de tamanho no máximo t.
Isso significa que para toda aresta $uv \in E(\overline{G})$, temos $u \in S$ ou $v \in S$.
A contrapositiva disso nos diz que se $u \notin S$ e $v \notin S$, então $uv \notin E(\overline{G})$, o que por sua vez implica em $uv \in E(G)$.
Logo, $V(G) \setminus S$ é uma clique em G.

Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER. Então $\overline{G}$ contém uma cobertura por vértices S de tamanho no máximo t.
Isso significa que para toda aresta $uv \in E(\overline{G})$, temos $u \in S$ ou $v \in S$.
A contrapositiva disso nos diz que se $u \notin S$ e $v \notin S$, então $uv \notin E(\overline{G})$, o que por sua vez implica em $uv \in E(G)$.
Logo, $V(G) \setminus S$ é uma clique em G.
Como $S \leq t = |V(G)| - k$, temos
$|V(G) \setminus S| = |V(G)| - |S| \geq |V(G)| - (|V(G)| - k) = k$.

Outro problema NP-completo: VERTEXCOVER

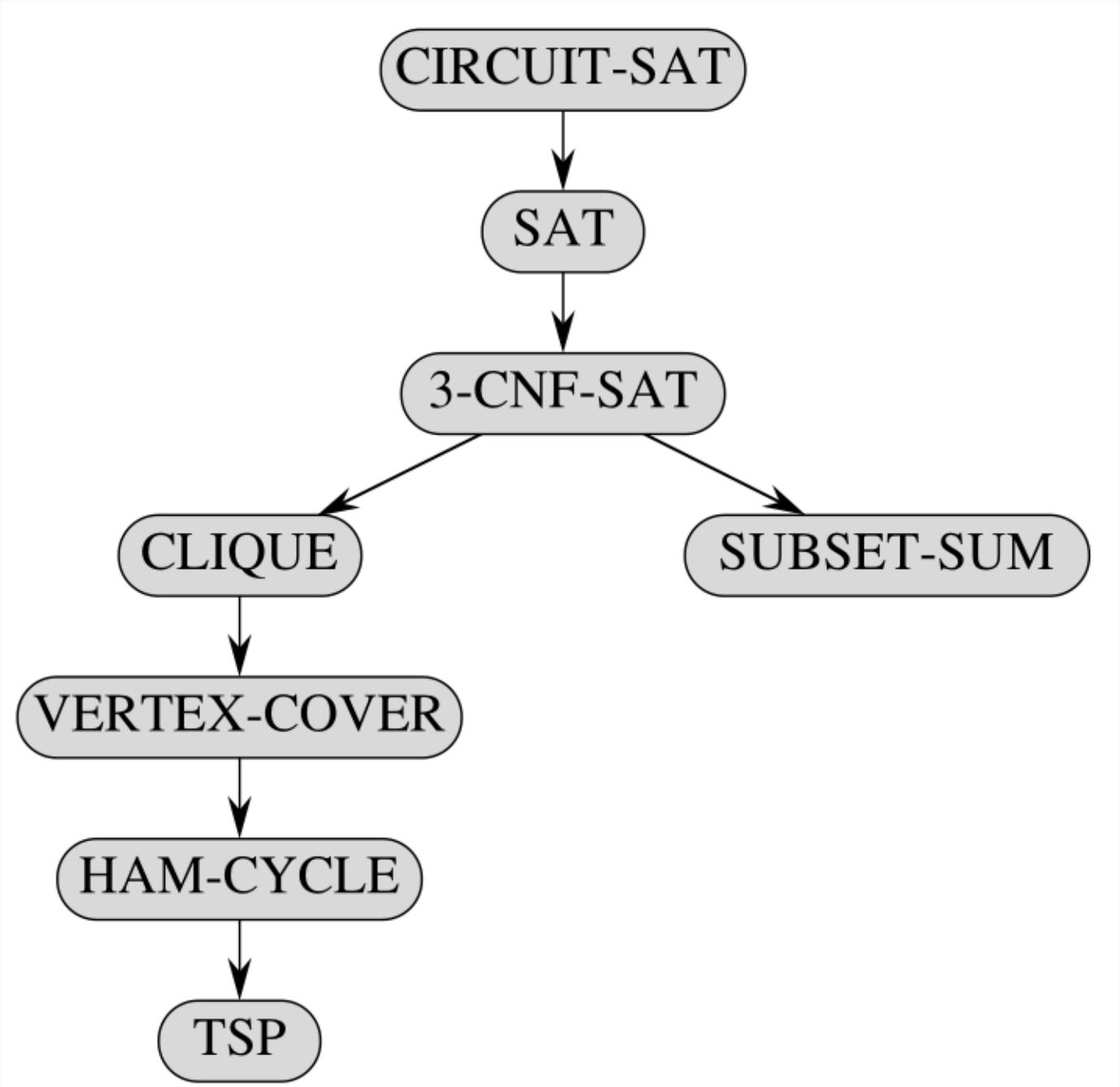
Teorema

VERTEXCOVER é NP-completo.

Demonstração (continuação)

Agora suponha que $\langle \overline{G}, t \rangle$ é `sim` para VERTEXCOVER. Então $\overline{G}$ contém uma cobertura por vértices S de tamanho no máximo t.
Isso significa que para toda aresta $uv \in E(\overline{G})$, temos $u \in S$ ou $v \in S$.
A contrapositiva disso nos diz que se $u \notin S$ e $v \notin S$, então $uv \notin E(\overline{G})$, o que por sua vez implica em $uv \in E(G)$.
Logo, $V(G) \setminus S$ é uma clique em G.
Como $S \leq t = |V(G)| - k$, temos $|V(G) \setminus S| = |V(G)| - |S| \geq |V(G)| - (|V(G)| - k) = k$. Então $\langle G, k \rangle$ é `sim` para CLIQUE. C.Q.D.

Classe NP-completo



Outro problema NP-completo: CICLOHAMILT

Um ciclo Hamiltoniano é um ciclo que passa por todos os vértices; isto é, é uma sequência $(v_1, \ldots, v_n, v_1 = v_{n+1})$ dos vértices tais que $v_i v_{i+1} \in E(G)$ para todo $1 \leq i \leq n$.

Problema: CICLOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe ciclo hamiltoniano em G?

Outro problema NP-completo: CICLOHAMILT

Um ciclo Hamiltoniano é um ciclo que passa por todos os vértices; isto é, é uma sequência $(v_1, \ldots, v_n, v_1 = v_{n+1})$ dos vértices tais que $v_i v_{i+1} \in E(G)$ para todo $1 \leq i \leq n$.

Problema: CICLOHAMILT

Entrada: $\langle G \rangle$, onde G é um grafo.
Decisão: existe ciclo hamiltoniano em G?

Teorema

CICLOHAMILT é NP-completo.

Demonstração

Primeiro mostramos que CICLOHAMILT está em NP e depois fazemos uma redução polinomial de VERTEXCOVER para CICLOHAMILT.

Outro problema NP-completo: CICLOHAMILT

Teorema

CICLOHAMILT é NP-completo.

Demonstração (continuação)

CLICLOHAMILT está em NP pois, dada uma sequência de vértices, podemos verificar se todos os vértice estão nela e se todo par de vértices consecutivos tem uma aresta entre eles no grafo. Isso pode ser feito apenas olhando a vizinhança de cada vértice da sequência e, portanto, leva tempo $O(|V(G)||E(G)|)$.

Outro problema NP-completo: CICLOHAMILT

Teorema

CICLOHAMILT é NP-completo.

Demonstração (continuação)

Considere um grafo G e um valor k inteiro, instância para VERTEXCOVER. Vamos construir um grafo G' da seguinte forma.

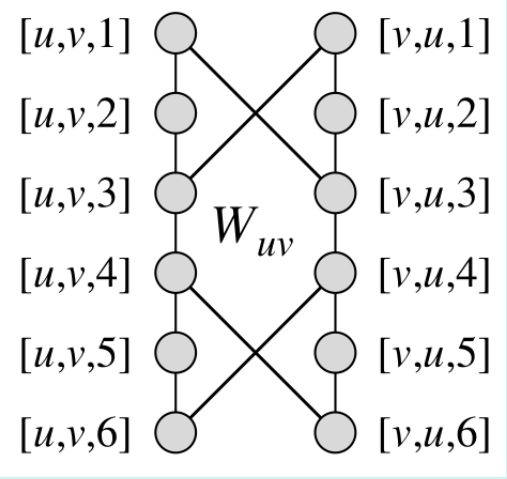
Outro problema NP-completo: CICLOHAMILT

Teorema

CICLOHAMILT é NP-completo.

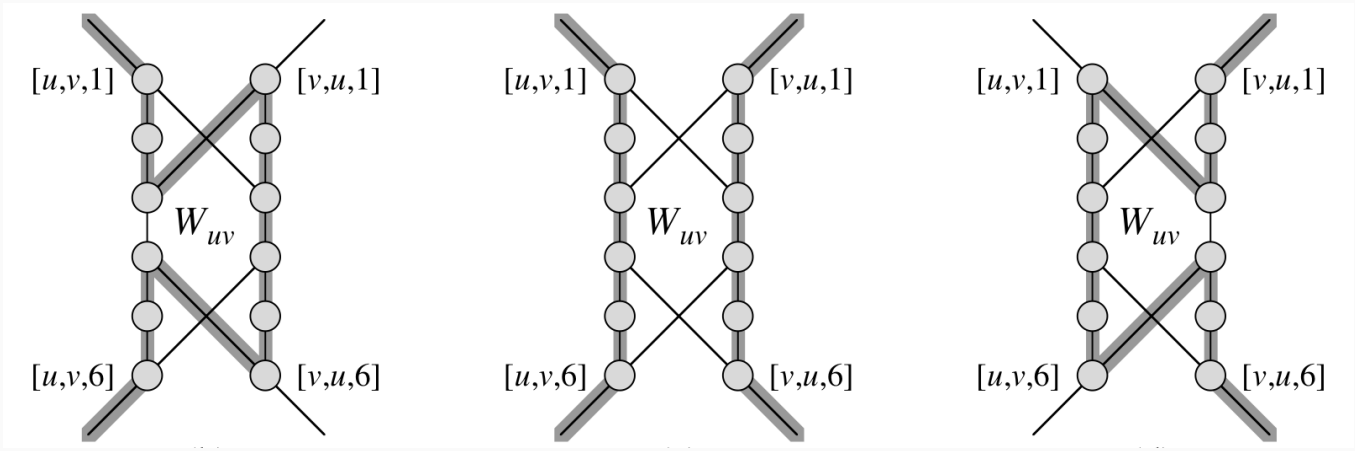
Demonstração (continuação)

Considere um grafo G e um valor k inteiro, instância para VERTEXCOVER. Vamos construir um grafo G' da seguinte forma. Para cada aresta $uv \in E(G)$, construa o seguinte grafo W_{uv} (widget) que tem 12 vértices e 14 arestas e inclua-o em G':



Outro problema NP-completo: CICLOHAMILT

Intuição: apenas os vértices $[u, v, 1]$, $[u, v, 6]$, $[v, u, 1]$ e $[v, u, 6]$ terão arestas incidentes de fora do widget.



Outro problema NP-completo: CICLOHAMILT

Teorema

CICLOHAMILT é NP-completo.

Demonstração (continuação)

Dado um vértice $u \in V(G)$, sejam $v_1, \ldots, v_{d(u)}$ seus vizinhos.
Adicione em G' as arestas $\{\{[u, v_i, 6], [u, v_{i+1}, 1]\} : 1 \leq i < d(u)\}$.

Intuição: note que isso cria um caminho em G' passando por todos os widgets que correspondem às arestas incidentes ao vértice u.

Outro problema NP-completo: CICLOHAMILT

Por fim, insira em G' outros k vértices $s_1, \ldots, s_k$, denominados seletores. Inclua as arestas $\{\{s_j, [u, v_1, 1]\} : u \in V(G) \text{ e } 1 \leq j \leq k\} \cup$ $\{\{s_j, [u, v_{d(u)}, 6]\} : u \in V(G) \text{ e } 1 \leq j \leq k\}$.

Intuição: cada seletor vai indicar um vértice para a cobertura; veja que apenas duas arestas do ciclo Hamiltoniano podem passar por cada s_j.

Outro problema NP-completo: CICLOHAMILT

Por fim, insira em G' outros k vértices $s_1, \ldots, s_k$, denominados seletores. Inclua as arestas $\{\{s_j, [u, v_1, 1]\}\colon u \in V(G) \text{ e } 1 \leq j \leq k\} \cup$ $\{\{s_j, [u, v_{d(u)}, 6]\}\colon u \in V(G) \text{ e } 1 \leq j \leq k\}$.

Intuição: cada seletor vai indicar um vértice para a cobertura; veja que apenas duas arestas do ciclo Hamiltoniano podem passar por cada s_j.

A construção levou tempo polinomial no tamanho de G, pois, sendo $n = |V(G)|$ e $m = |E(G)$, note que G' tem $12m + k \leq 12m + n$ vértices e $16m + (2k - 1)n \leq 16m + (2n - 1)n$ arestas.

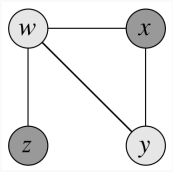
Outro problema NP-completo: CICLOHAMILT

Por fim, insira em G' outros k vértices $s_1, \ldots, s_k$, denominados seletores. Inclua as arestas $\{\{s_j, [u, v_1, 1]\} : u \in V(G) \text{ e } 1 \leq j \leq k\} \cup \{\{s_j, [u, v_{d(u)}, 6]\} : u \in V(G) \text{ e } 1 \leq j \leq k\}$.

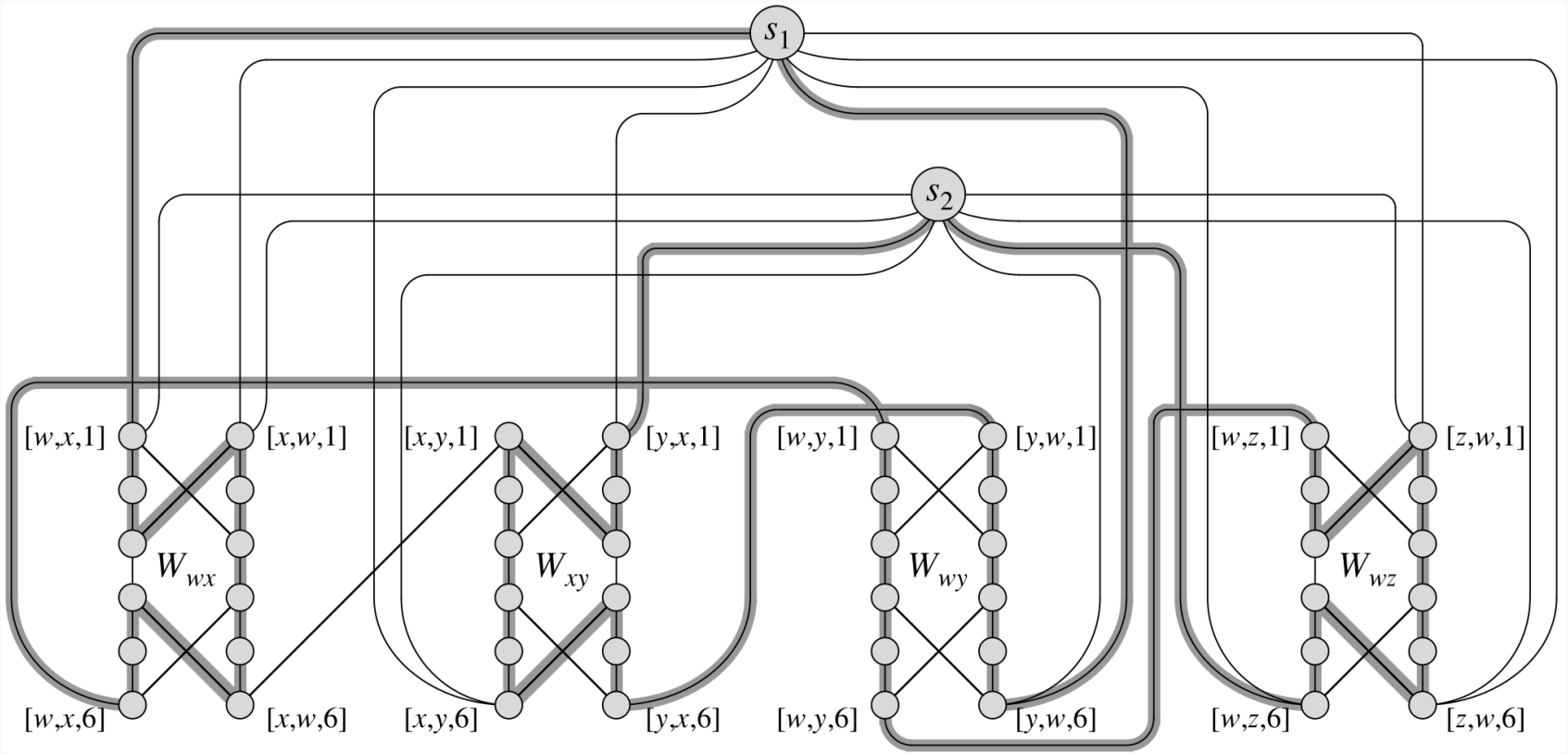
Intuição: cada seletor vai indicar um vértice para a cobertura; veja que apenas duas arestas do ciclo Hamiltoniano podem passar por cada s_j.

A construção levou tempo polinomial no tamanho de G, pois, sendo $n = |V(G)|$ e $m = |E(G)$, note que G' tem $12m + k \leq 12m + n$ vértices e $16m + (2k - 1)n \leq 16m + (2n - 1)n$ arestas.
Resta mostrar que $\langle G, k\rangle \in$ VERTEXCOVER se e somente se $\langle G'\rangle \in$ CICLOHAMILT. $\star$ C.Q.D.

Outro problema NP-completo: CICLOHAMILT

Para o grafo G com 4 vértices 

, o grafo G' correspondente é



Lembrete

Dizer que um problema está em NP não é argumento para sua intratabilidade.

Dizer que ele é NP-completo/NP-difícil, sim.

O problema do Caixeiro Viajante - Exercício

Outra linguagem NP-completa: TSP

Problema: TSP

Entrada: $\langle G, w, k \rangle$, onde G é um grafo, $w\colon E(G) \to \mathbb{R}$ e $k \in \mathbb{R}$.
Decisão: existe ciclo hamiltoniano em G com peso $\leq k$?

Outra linguagem NP-completa: TSP

Problema: TSP

Entrada: $\langle G, w, k \rangle$, onde G é um grafo, $w\colon E(G) \to \mathbb{R}$ e $k \in \mathbb{R}$.
Decisão: existe ciclo hamiltoniano em G com peso $\leq k$?

Teorema

TSP é NP-completo.

Demonstração

Primeiro mostramos que TSP está em NP e depois faremos uma redução polinomial de CICLOHAMILT para TSP.

TSP está em NP pois ??? (Exercício)

Outra linguagem NP-completa: CICLOHAMILT

Teorema

TSP é NP-completo.

Demonstração (continuação)

Considere um grafo G, instância para CICLOHAMILT.

Outra linguagem NP-completa: CICLOHAMILT

Teorema

TSP é NP-completo.

Demonstração (continuação)

Considere um grafo G, instância para CICLOHAMILT.

Crie um novo grafo G' completo, com os mesmos vértices de G. Atribua os custos da seguinte forma: $w(e) = 0$ se $e \in E(G)$ e $w(e) = 1$ se $e \notin E(G)$. Por fim, tome $k = 0$.

Outra linguagem NP-completa: CICLOHAMILT

Teorema

TSP é NP-completo.

Demonstração (continuação)

Considere um grafo G, instância para CICLOHAMILT.

Crie um novo grafo G' completo, com os mesmos vértices de G. Atribua os custos da seguinte forma: $w(e) = 0$ se $e \in E(G)$ e $w(e) = 1$ se $e \notin E(G)$. Por fim, tome $k = 0$.

Resta provar que $\langle G \rangle$ é `sim` para CICLOHAMILT se e somente se $\langle G', w, k \rangle$ é `sim` para TSP. (Exercício)

Usando NP-completude para provar outras coisas (EXTRA - CURIOSIDADE)

Teorema

Não existe algoritmo eficiente que resolva o problema CAMINHOTODOPAR sobre $\langle D, w \rangle$ quando há ciclo negativo em D, a menos que $P = NP$.

Demonstração

Primeiro note que há uma redução polinomial de CAMINHOLONGOOPT para CAMINHOTODOPAR.
Seja $\langle D \rangle$ uma instância para CAMINHOLONGOOPT.
Faça $w(e) = -1$ para todo arco $e \in E(D)$.
Note que $\langle D, w \rangle$ é instância para CAMINHOTODOPAR.
Se P é o caminho de maior comprimento em D, então os vértices extremos de P têm a menor distância considerando a função w.
Da mesma forma, se u, v são pares de vértices com a menor distância considerando a função w, então um uv-caminho mínimo é um caminho de maior comprimento em D.

Teorema

Não existe algoritmo eficiente que resolva o problema CAMINHOTODOPAR sobre $\langle D, w \rangle$ quando há ciclo negativo em D, a menos que $P = NP$.

Demonstração (continuação)

Se D não tem ciclos, então encontrar o caminho mais longo em D pode ser feito em tempo polinomial. Logo, considere que D tem ciclos.
Suponha agora que existe um algoritmo eficiente ALG que é capaz de resolver o problema CAMINHOTODOPAR sobre qualquer digrafo que possua ciclos negativos.
Em particular, $ALG(D, w)$ encontrará um par x, y de vértices cuja distância é a menor dentre as distâncias de todos os pares de vértices de D.
Com isso, podemos usar ALG para resolver o problema CAMINHOLONGOOPT, que é NP-difícil.