Algoritmos

Pedro Hokama

Fontes

- [*clrs*] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest e Clifford Stein.
- [*timr*] Algorithms Illuminated Series, Tim Roughgarden
- Desmistificando Algoritmos, Thomas H. Cormen.
- Algoritmos, Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani
- Stanford Algorithms
  https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V
  https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6
- Conjunto de Slides dos Professores Cid C. de Souza, Cândida N. da Silva, Orlando Lee, Pedro J. de Rezende
- Conjunto de Slides do Professores Cid C. de Souza para a disciplina MO420

Qualquer erro é de minha responsabilidade.

CLICK

Problema do CLICK

Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe um subgrafo G' induzido de G que tenha k vértices e seja completo.

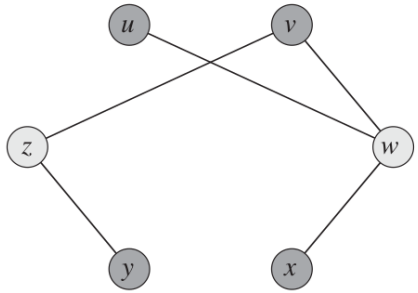
Teorema

CLICK é NP-completo

- Provado com 3-CNF-SAT $\leq_p$ CLICK

Cobertura por Vértices

- Dado um grafo não direcionado $G = (V, E)$ uma cobertura por vértices de G é um subconjunto $V' \subseteq V$ tal que para toda aresta $(u, v) \in E$, pelo menos um entre u e v deve estar em V'.
- Dizemos que um vértice em V' cobre todas as arestas adjacentes a ele. E em uma cobertura por vértices todas as arestas devem ser cobertas.
- O **tamanho** de uma cobertura por vértices é o número de vértices que contêm.



Problema da Cobertura por Vértices - VERTEX-COVER

Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe uma cobertura por vértices $V' \subseteq V$ de tamanho k.

Teorema

VERTEX-COVER é NP-completo

Lema

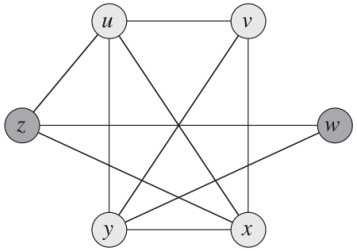
VERTEX-COVER $\in$ *NP*

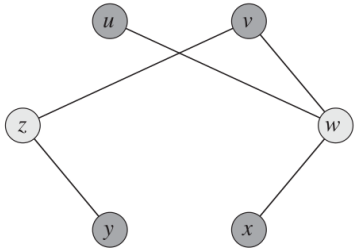
- Utilizando a própria cobertura V' como certificado, podemos verificar facilmente, em tempo polinomial, se $|V'| = k$ e se toda aresta (u, v), u ou v está em V'.

Lema

VERTEX-COVER $\in$ *NP-Difícil*

- Vamos mostrar que CLICK $\leq_p$ VERTEX-COVER.
- Definição: O **Complemento** de um grafo $G = (V, E)$ denotado por $\overline{G} = (V, \overline{E})$, em que, $\overline{E} = \{(u, v) : u, v \in V, u \neq v \text{ e } (u, v) \notin E\}$. Ou seja, o complemento de G é um grafo com os mesmos vértices e exatamente as arestas que não estão em G.
- A redução consiste em dada uma entrada $< G, k >$ do problema da CLICK. Calcular o complemento $\overline{G}$, o que pode ser feito em tempo polinomial.
- E então usar $\overline{G}$ como entrada para o problema do VERTEX-COVER, procurando uma cobertura de tamanho $|V| - k$.





Lema

G tem uma CLICK de tamanho k, se e somente se, $\overline{G}$ *tem uma cobertura por vértices de tamanho* $|V| - k$

- ($\rightarrow$) Se G tem uma CLICK C com k vértices, então $V \setminus C$ é uma cobertura em $\overline{G}$.
- Seja (u, v) qualquer aresta em $\overline{E}$, então (u, v) não está em G e portanto u e v não podem estar em C simultaneamente.
- Logo, pelo menos um deles está em $V \setminus C$ e portanto (u, v) está coberta.
- ($\leftarrow$) Se $\overline{G}$ tem uma cobertura por vértices $V' \subseteq V$ em que $|V'| = |V| - k$, então para todo $u, v \in V$ se $(u, v) \in \overline{E}$ então ou $u \in V'$ ou $v \in V'$ ou ambos. Pela contrapositiva se nenhum dos dois está em V' significa que (u, v) está em E e portanto $V - V'$ é uma CLICK, e tem tamanho $|V| - |V'| = k$
- Portanto mostramos um redução CLICK $\leq_p$ VERTEX-COVER. $\square$

A questão P vs NP

- Seria $P = NP$?
- Bastaria mostrar 1 algoritmo com tempo de execução polinomial para 1 problema NP-completo.
- Conjectura-se (fortemente) que $P \neq NP$.
- Mas também não foi provado.

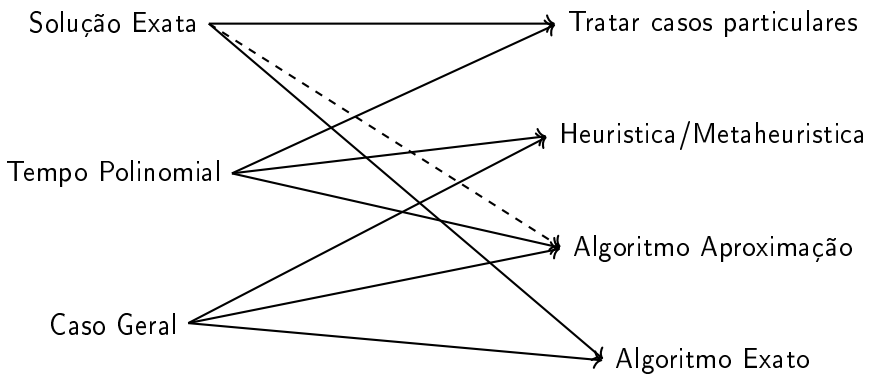
O nome NP

- O nome NP deriva de Non-deterministic Polynomial Time
- São problemas que podem ser resolvidos em tempo polinomial em uma máquina de Turing não determinística.
- Informalmente, considere uma máquina que conseguisse testar todas as soluções simultaneamente.

Problemas Intratáveis

- O que podemos fazer se um problema é NP-completo? (sem ser desistir)
- Na verdade muita coisa pode ser feita, e é disso que trataremos nessa disciplina.

Se $P \neq NP$ não conseguiremos para um problema NP-Difícil:



Casos Especiais

- Muitas vezes um problema geral pode ser *NP*-completo mas alguns casos especiais podem ser mais fáceis de resolver
  1. Encontrar um conjunto independente máximo é *NP*-completo. Mas no grafo caminho é fácil.
  2. O problema da Mochila binária é *NP*-completo, mas o da mochila fracionária é fácil.
  3. Na mochila binária, se a capacidade for polinomial no número de itens o problema também é fácil
  4. Satisfabilidade de fórmulas booleanas é *NP*-completo, mas o 2-CNF-SAT é fácil

Heurísticas

- Podemos abrir mão de encontrar a solução ótima, e tentar encontrar uma solução boa o bastante.
- Normalmente heurísticas são rápidas.
- Geralmente não possuem garantias do quão próximas estão da solução ótima

Algoritmos Aproximados

- Podemos abrir mão de encontrar a solução ótima, e tentar encontrar uma solução boa o bastante.
- Executam em tempo polinomial.
- Possuem garantias do quão próximas estão da solução ótima.
- Por exemplo: A solução de um algoritmo aproximado vai estar no máximo a um fator α da solução ótima.

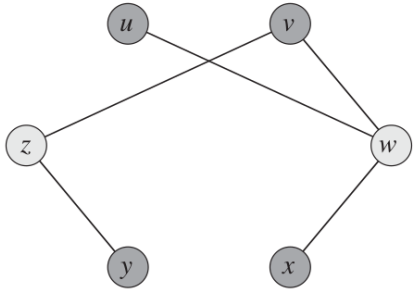
Algoritmos Exatos

- Busca a solução ótima.
- Desiste do tempo polinomial.
- A ideia é reduzir ao máximo a complexidade, e aumentar ao máximo o tamanho das instâncias que conseguimos resolver.
  - Branch-and-Bound
  - Programação Linear Inteira
  - Programação por restrições

Cobertura por Vértices

VERTEX-COVER

Dado um grafo não orientado $G = (V, E)$ com n vértice e m arestas, e um inteiro k decidir se existe uma cobertura por vértices $V' \subseteq V$ de tamanho k.



Qual é o tamanho de uma cobertura por vértices de tamanho mínimo em um grafo estrela com n vértices e em um grafo clique de tamanho n.

a) 1 e $n-1$
b) 1 e n
c) 2 e $n-1$
d) $n-1$ e n

- Uma solução força bruta:
- Considerando que k seja pequeno em relação a n podemos tentar todos os conjuntos com k vértices.
- São $\binom{n}{k}$ conjuntos.
- Se $k <<< n$ então a complexidade do algoritmo é $\approx \Theta(n^k)$.
- Podemos fazer melhor?

Teorema

Dada uma aresta (u, v),
$G_u = G$ deletando u e todas as suas arestas adjacentes, e
$G_v = G$ deletando v e todas as suas arestas adjacentes.

G tem uma cobertura de tamanho k, se e somente se
G_u ou G_v tem uma cobertura de tamanho $k - 1$.

- ($\rightarrow$) Suponha que G tem uma cobertura V' de tamanho k. Como a aresta (u, v) precisa estar coberta, pelo menos um dos seus extremos tem que estar em V'.
- Sem perda de generalidade, suponha que $u \in V'$.
- O vértice u cobre apenas as arestas adjacentes a u, e portanto todas as demais arestas devem estar cobertas pelos $k - 1$ vértices restantes de V', e portanto
- $V \backslash \{u\}$ é uma cobertura para G_u e tem $k - 1$ vértices
- ($\leftarrow$) Exercício.

Algoritmo 1: BuscaCobertura

Entrada: Um grafo $G(V, E)$ e um inteiro k
Saída: Verdadeiro se G contêm uma cobertura de tamanho k

```
1 se |E| > 0 e k == 0 então devolve Falso
2 se |E| == 0 então devolve Verdadeiro
3 Escolhe uma aresta qualquer (u, v) ∈ E
4 Cria G_u = G sem u e suas arestas adjacentes
5 Cria G_v = G sem v e suas arestas adjacentes
6 se BuscaCobertura(G_u, k − 1) então devolve Verdadeiro
7 se BuscaCobertura(G_v, k − 1) então devolve Verdadeiro
8 devolve Falso
```


Complexidade de BuscaCobertura:

- A cada chamada recursiva, fazemos outras 2.
- A profundidade da recursão é no máximo k.
- Portanto o número de chamadas recursivas é no máximo 2^k.
- Cada chamada recursiva leva tempo $O(m)$ para criar G_u e G_v.
- Portanto a complexidade total é $O(2^k m)$, portanto exponencial. (Claro que é).
- Ainda muito melhor que o $\Theta(n^k)$ da força bruta.