Métodos Exatos

Pedro Hokama

Fontes

- Curso Discrete Optimization no Coursera - Prof. Dr. Pascal Van Hentenryck
- [*clrs*] Algoritmos: Teoria e Prática (Terceira Edição) Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest e Clifford Stein.
- [*timr*] Algorithms Illuminated Series, Tim Roughgarden
- Pesquisa Operacional, Arenales, M. N., Armentano, V. A., Morabito Neto, R., e Yanasse, H. H. (2015)
- Algoritmos, Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani
- Stanford Algorithms
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V`
  `https://www.youtube.com/playlist?list=PLXFMmlk03Dt5EMI2s2WQBsLsZl7A5HEK6`
- Introdução à Otimização Combinatória, Flávio K. Miyazawa e Cid C. de Souza.
- Otimização Combinatória, da Carla Negri Lintzmayer do CMCC, UFABC

Qualquer erro é de minha responsabilidade.

3-CNF-SAT

- Uma formula booleana é composta
  - de variáveis booleanas $x_1, x_2, \ldots, x_n$,
  - de conectivos $\neg$ (NOT), $\vee$ (OR), $\wedge$ (AND),
  - Parenteses.
- Uma variável ou sua negação são chamadas **literais**.
- Uma fórmula booleana está na forma normal 3-conjuntiva se
  - está dividido em clausuras
  - cada clausura tem 3 literais conectas por ORs
  - as clausuras estão conectadas por ANDs.

$$(x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_3 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee \neg x_4)$$

3-CNF-SAT

3-CNF-SAT

Dado uma fórmula booleana na forma normal 3-conjuntiva, decidir se existe alguma atribuição das variáveis que torna a formula verdadeira.

Teorema

O 3-CNF-SAT é NP-Completo

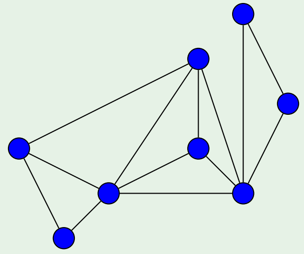
Prova: Cap. 34 do CLRS.

Problema do CLICK

Problema do CLICK

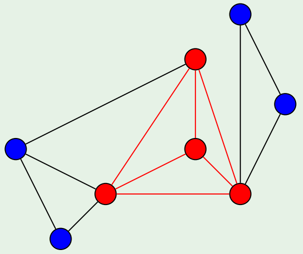
Dado um grafo não orientado $G = (V, E)$, e um inteiro k decidir se existe um subgrafo G' induzido de G que tenha k vértices e seja completo.

Exemplo



Existe uma click de tamanho 4?

Exemplo



Existe uma click de tamanho 4?

O Problema do CLICK

Será que CLICK é NP-Completo?

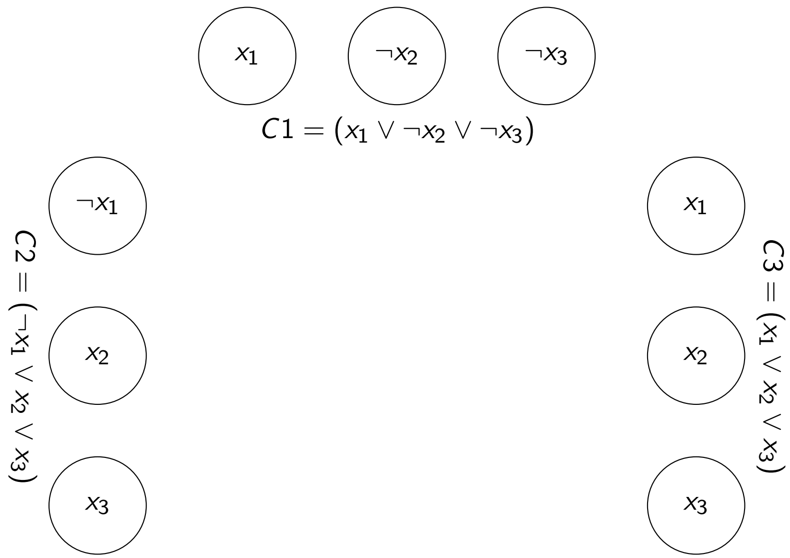
- CLICK $\in$ NP, basta apresentar o conjunto de vértices
- Vamos reduzir o 3-CNF-SAT para CLICK
  - Devemos converter qualquer instancia do 3-CNF-SAT para o problema do CLICK
  - Essa redução deve ter tempo polinomial
  - Aqui vamos mostrar o algoritmo da redução em um exemplo, mas é necessário garantir que funciona para qualquer instância.

3-CNF-SAT $\leq_p$ CLICK

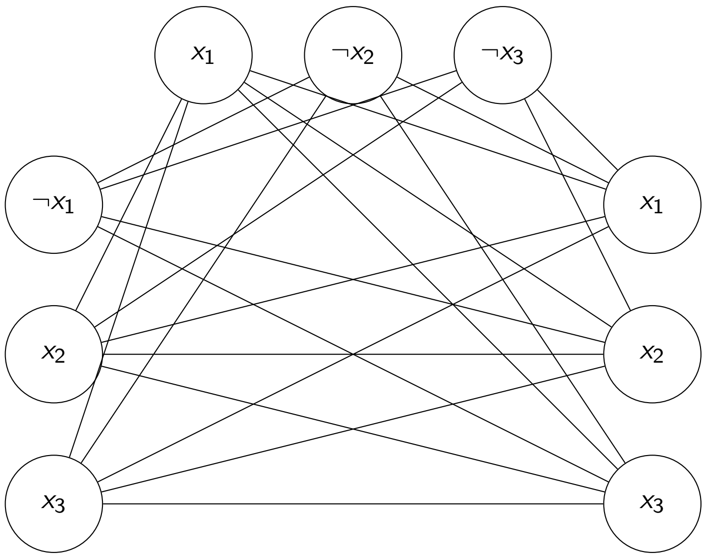
Considere uma formula booleana com k clausulas na forma normal 3-conjuntiva.

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

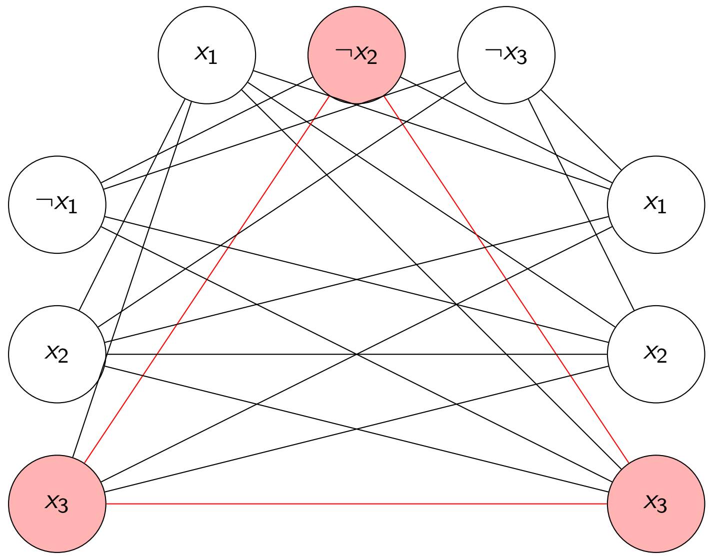
$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$



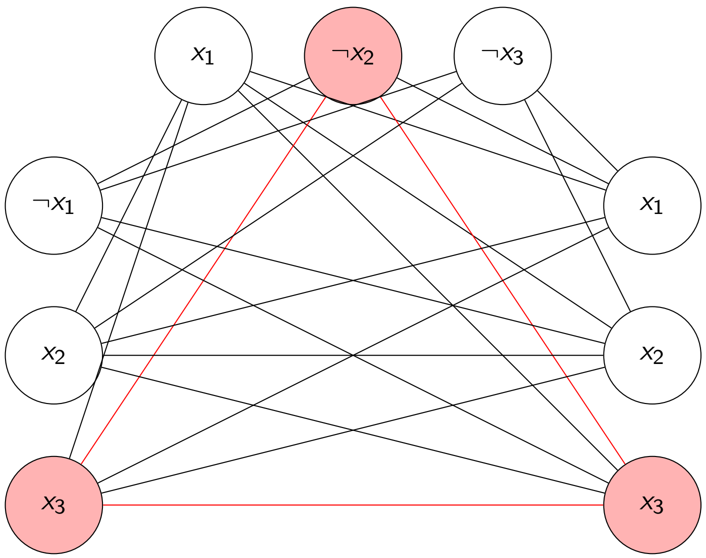
$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$



$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$$

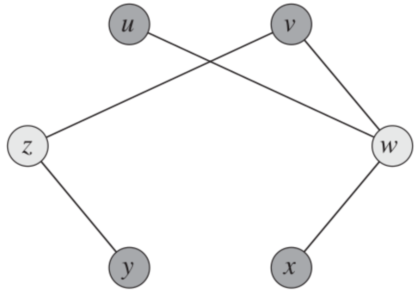


$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_3)$



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

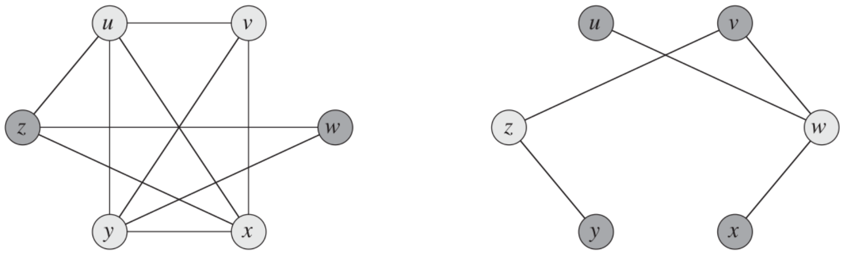
VERTEX-COVER $\in$ *NP*

- 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.

Millennium Problems

Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

Riemann Hypothesis

The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

P vs NP Problem

If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation

This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.

Hodge Conjecture

The answer to this conjecture determines how much of the topology of the solution set of a system of algebraic equations can be defined in terms of further algebraic equations. The Hodge conjecture is known in certain special cases, e.g., when the solution set has dimension less than four. But in dimension four it is unknown.

Poincaré Conjecture

In 1904 the French mathematician Henri Poincaré asked if the three dimensional sphere is characterized as the unique simply connected three manifold. This question, the Poincaré conjecture, was a special case of Thurston's geometrization conjecture. Perelman's proof tells us that every three manifold is built from a set of standard pieces, each with one of eight well-understood geometries.

Birch and Swinnerton-Dyer Conjecture

Supported by much experimental evidence, this conjecture relates the number of points on an elliptic curve mod p to the rank of the group of rational points. Elliptic curves, defined by cubic equations in two variables, are fundamental mathematical objects that arise in many areas: Wiles' proof of the Fermat Conjecture, factorization of numbers into primes, and cryptography, to name three.

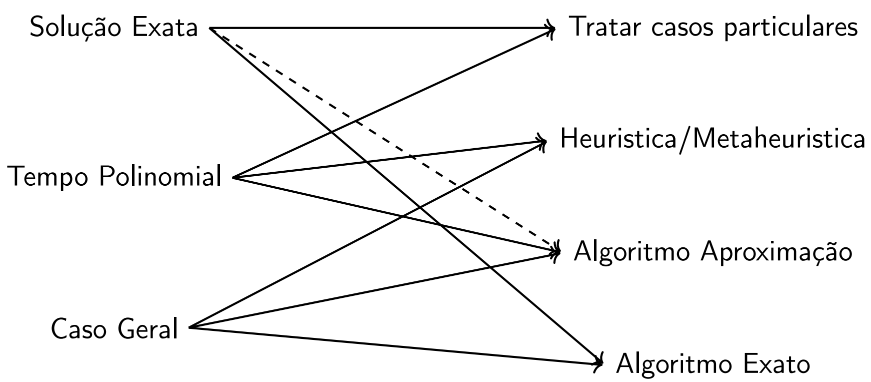
O nome NP

- 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