Problemas NP-Completude

Exemplos de Demonstrações de Problemas NP-Completos.

Problema Clique Máximo do Grafo

- **Um clique em um grafo não direcionado G = (V, E) é um subconjunto V' ⊂ V de vértices, cada par dos quais é conectado por uma aresta em E.**
- **Em outras palavras, um clique é um subgrafo completo de G. O tamanho de um clique é o número de vértices que ele contém.**
- **O problema do clique é o problema de otimização de encontrar um clique de tamanho máximo em um grafo.**

Problema Clique Máximo do Grafo

- **Como um problema de decisão, nós perguntamos simplesmente se um clique de um dado tamanho k existe no grafo. A definição formal é:**

  CLIQUE={<G,k>:G é um grafo contendo um clique de tamanho k}

- **Para mostrar que um problema é NP-Completo precisamos:**
  - que ele é NP ($L \in NP$), e
  - que ele é redutível polinomialmente a outro problema NP ($L' \leq_p L$ para todo $L' \in NP$)

Problema Clique Máximo do Grafo

Teorema:

O problema clique máximo do grafo é NP-Completo.

Prova:

- CLIQUE Є NP:
  - para um grafo G=(V,E), vamos usar o conjunto V' ⊂ V de vértices no certifica de clique para G.
  - podemos verificar se V' é um clique em tempo polinomial verificando se, para cada par u,v Є V', a aresta (u,v) pertence a E.

Problema Clique Máximo do Grafo

- 3-CNF-SAT $\leq_P$ CLIQUE:
  - O algoritmo de redução começa com uma instância de 3-CNF-SAT. Seja a fórmula booleana 3-CFN com k cláusulas

    $\Phi = C_1 \wedge C_2 \wedge \ldots \wedge C_k$
  - onde cada cláusula C_r, para r=1, 2, .., k, tem 3 literais l_1^r, l_2^r e l_3^r.
  - Devemos construir um grafo G tal que Φ é satisfeita se e somente se G tem um clique de tamanho k.

Problema Clique Máximo do Grafo

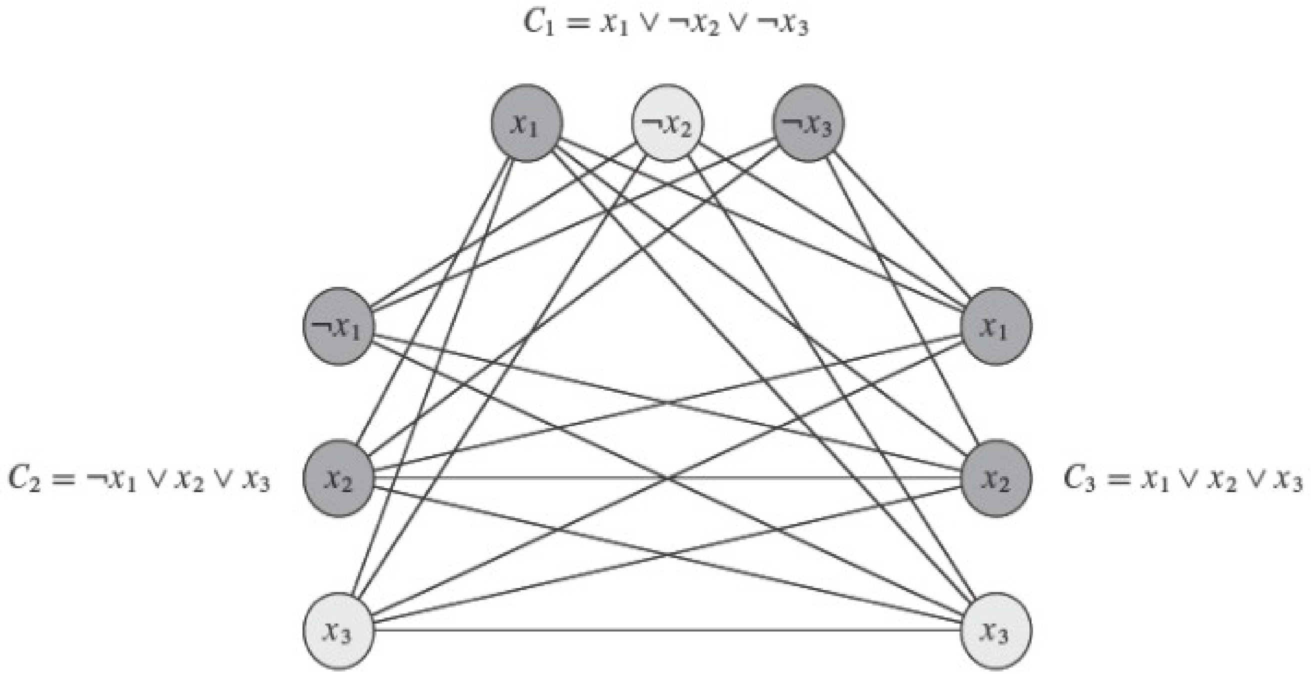
- Este grafo G=(V,E) deve ser construído da seguinte forma:
  - Para cada cláusula $C_r = (l_1^r \vee l_2^r \vee l_3^r)$ em Φ criamos 3 vértices v_1^r , v_2^r , v_3^r e colocamos arestas entre dois vértices v_i^r e v_j^s se ambas as seguintes condições forem válidas:
    - v_i^r e v_j^s estão em diferentes triplas, ou seja $i \neq j$, e
    - Seus correspondentes literais são consistentes, isto é l_i^r não é negação de l_j^s .
- Podemos facilmente construir esse grafo em tempo polinomial a partir de Φ.

Problema Clique Máximo do Grafo

- **Como exemplo dessa construção, se temos**

  $\Phi = (x_1 \lor !x_2 \lor !x_3) \land (!x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3)$

- **Então o grafo G é:**



Problema Clique Máximo do Grafo

- **É preciso mostrar que a transformação de Φ em G é uma redução.**
  - Primeiro suponha que Φ tem uma atribuição satisfatória;
  - então cada cláusula C_r contem pelo menos um literal l_i^r com valor 1, e cada tal literal corresponde a um vértice v_i^r ,
  - Selecionar um literal "verdadeiro" de cada cláusula produz um conjunto V' de k vértices. Afirmamos que V' é um clique.
  - Para quaisquer dois vértices v_i^r, $v_j^s \in V'$, onde $r \neq s$, ambos correspondendo aos literais l_i^r , l_j^s mapeados para 1 pela atribuição satisfatória, e assim os literais não podem ser correspondentes. Assim, pela construção de G, a aresta (v_i^r, v_j^s) pertence a E.

Problema Clique Máximo do Grafo

- Por outro lado, suponha que G tenha um clique V' de tamanho k. Nenhuma aresta em G conecta vértices na mesma tripla, e então V' contém exatamente um vértice por tripla. Podemos atribuir 1 a cada literal l_i^r tal que $v_i^r \in V'$ sem medo de atribuir 1 a um literal e seu complemento, já que G não contém arestas entre literais inconsistentes. Cada cláusula é satisfeita, e então Φ é satisfeita. (Quaisquer variáveis que não correspondam a um vértice no clique podem ser definidas arbitrariamente.)

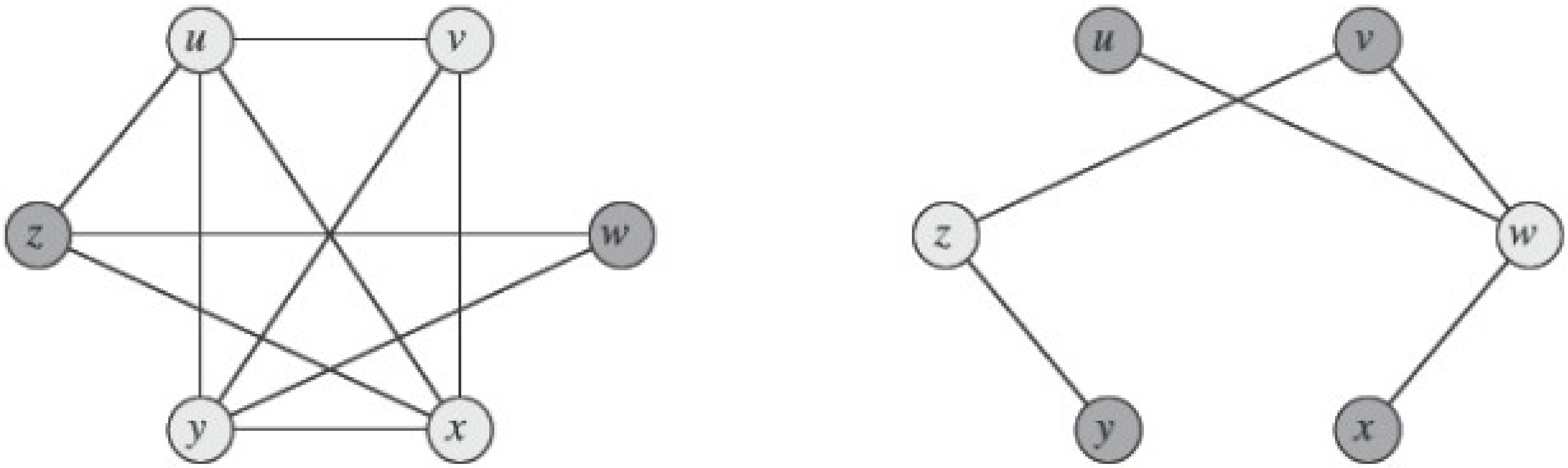
Problema Clique Máximo do Grafo

- No exemplo da Grafo G, uma atribuição satisfatória de Φ tem x_2=0 e x_3=1. Um clique correspondente de tamanho k = 3 consiste nos vértices correspondentes a !x_2 da primeira cláusula, x_3 da segunda cláusula e x_3 da terceira cláusula. Como o clique não contém vértices correspondentes a x_1 ou !x_1, podemos definir x_1 como 0 ou 1 nessa atribuição satisfatória.
- Observe que na prova do Teorema, reduzimos uma instância arbitrária de 3-CNF-SAT a uma instância de CLIQUE com uma estrutura particular. Você pode pensar que mostramos apenas que CLIQUE é NP-difícil em grafos nos quais os vértices são restritos a ocorrer em triplas e nos quais não há arestas entre vértices na mesma tripla. De fato, mostramos que CLIQUE é NP-difícil apenas neste caso restrito, mas esta prova é suficiente para mostrar que CLIQUE é NP-difícil em grafos gerais. Por quê? Se tivéssemos um algoritmo de tempo polinomial que resolvesse CLIQUE em grafos gerais, ele também resolveria CLIQUE em grafos restritos.

O problema da cobertura de vértices

- **Uma cobertura de vértices de um grafo não direcionado G=(V, E) é um subconjunto V' ⊂ V tal que se (u, v) ∈ E, então u ∈ V' ou v ∈ V' (ou ambos). Ou seja, cada vértice "cobre" suas arestas incidentes, e uma cobertura de vértices para G é um conjunto de vértices que cobre todas as arestas em E. O tamanho de uma cobertura de vértices é o número de vértices nela. Por exemplo, o grafo na Figura (b) a seguir tem uma cobertura de vértices {w,z} de tamanho 2.**

O problema da cobertura de vértices



(a) (b)

O problema da cobertura de vértices

- O problema da cobertura de vértices é encontrar uma cobertura de vértices de tamanho mínimo em um dado grafo. Reformulando esse problema de otimização como um problema de decisão, desejamos determinar se um grafo tem uma cobertura de vértices de um dado tamanho k. Como uma linguagem definimos

  VERTEX-COVER = { <G,k>: grafo G tem uma cobertura de vértices de tamanho k}

- Mostre que o problema de Cobertura de Vértices é NP-completo

O problema do caixeiro viajante

- No problema do caixeiro-viajante, que é intimamente relacionado ao problema do ciclo hamiltoniano, um vendedor deve visitar n cidades. Modelando o problema como um grafo completo com n vértices, podemos dizer que o vendedor deseja fazer um passeio, ou ciclo hamiltoniano, visitando cada cidade exatamente uma vez e terminando na cidade de onde começa.
- O vendedor incorre em um custo inteiro não negativo c(i,j) para viajar da cidade i para a cidade j, e o vendedor deseja fazer o passeio cujo custo total é mínimo, onde o custo total é a soma do custo individual ao longo das bordas do passeio

O problema do caixeiro viajante

- Por exemplo, na figura abaixo o passeio de custo mínimo é (u, w, v, x, u) com 7.
- A linguagem formal para o correspondente problema de decisão é

  TSP = { <G,c,k>: G=(V,E) é um grafo completo, c é uma função VxV→Z, k ∈ Z, e G tem um passeio do caixeiro viajante com custo no máximo k.}

- **Mostre que esse problema é NP-Completo.**