1. Cite dois problemas conhecidos, um de otimização e outro de decisão, e procure aplicações práticas desses problemas, provando se são P, NP ou NP-Completo.

1.1 Problema da Cobertura de Vértices (Otimização)

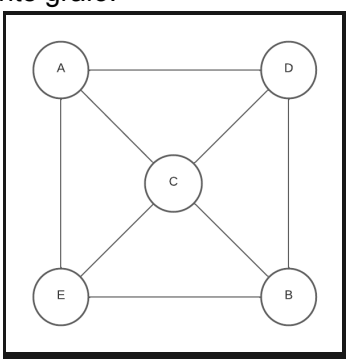
Serão necessárias duas etapas, sendo a primeira referente a demonstrar que o problema se encontra na classe NP apresentando um algoritmo não-determinista que execute a solução do problema em tempo polinomial. Mas há também uma outra forma de fazer a demonstração, que seria encontrar um algoritmo determinista polinomial que servirá de verificação para uma solução X válida. Já a segunda etapa diz respeito à prova de que o problema de Cobertura de Vértices está em NP-Completo reduzindo um problema já existente em um conjunto NP-Completo ao problema em questão CV (Cobertura de Vértices) em tempo polinomial.

Primeiramente mostra-se que o problema do CLIQUE pertence à classe NP, que no caso, será através da utilização de um algoritmo determinístico que faz a verificação da solução em tempo polinomial.

```
verify(G, S, k) {
  if (|S| for menor ou igual a k) {
        para toda e ∈ E //e é uma aresta do conjunto E de arestas.{
            remova aresta de aresta de G.
        }
        para todo v ∈ V {
            if (grau[v] ≠ 0) {
            return "Não existe cobertura de tamanho menor ou igual a k.";
            }
       }
     return true;
   }
  else if{
     return false;
  }
```

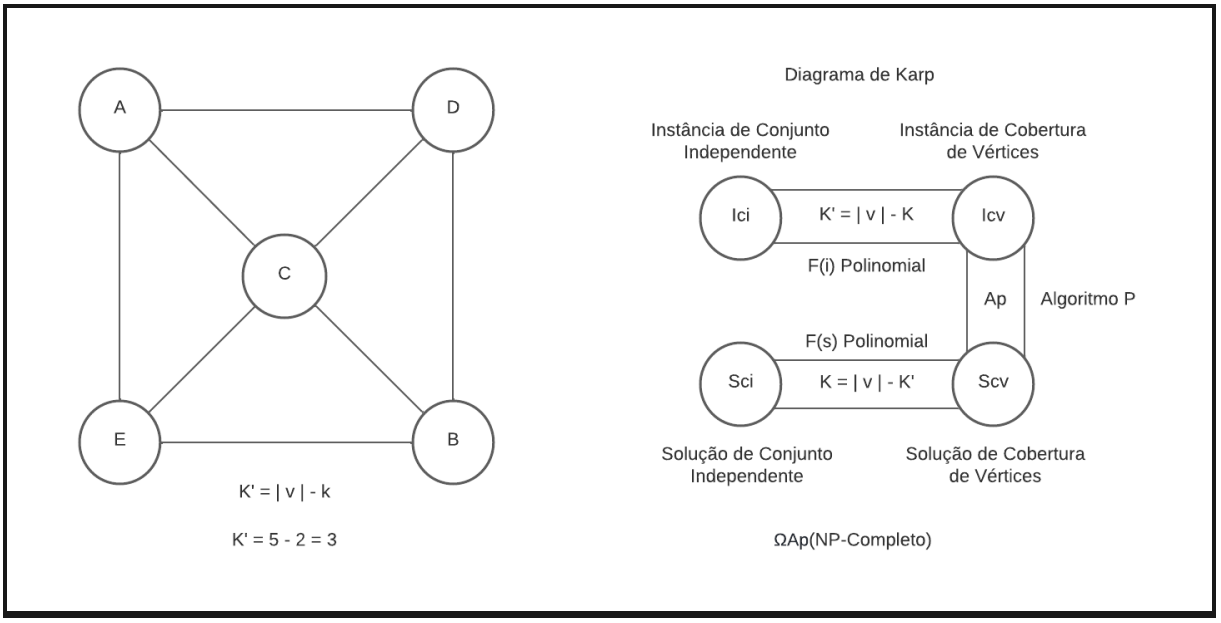
Agora temos que mostrar que a Cobertura de Vértices pertence à classe NP-Completo, logo teremos que fazer a redução do problema de conjunto independente ao problema CV com tempo polinomial.

Levando em consideração o seguinte grafo:



Podemos ter os seguintes problemas independentes: {A, B}, {C, D}
E também as seguintes coberturas de vértices: {A, C, B}, {E, C, D}

Exemplificação do processo a partir de um diagrama de Karp:



Sendo Ici o conjunto de instâncias referente ao conjunto independente, haverá uma função injetora F(i) que irá reduzir as instâncias em tempo polinomial referente às instâncias específicas da Cobertura de Vértices (Lcv)

A função F(i) reflete a seguinte ideia: Sendo um grafo X (o que foi representado na figura acima), caso exista um Conjunto Independente de tamanho k, essa função irá gerar um outro grafo igual ao grafo X, o qual sua Cobertura de vértices será k = | v | - k.

Ao chegar na outra parte do diagrama de Karp, temos a representação do algoritmo P (Ap) que irá solucionar a instância do problema Lcv e irá retornar os vértices que serão a solução do problema de Cobertura de Vértices, apresentando uma resposta SIM ou NÃO, ou até mesmo o valor de K'.

Por fim, sabendo que a cota inferior do problema de Conjunto Independente é Ω(NP-Completo) o algoritmo Ap não tem como ter sua cota inferior que Ω(NP-Completo)

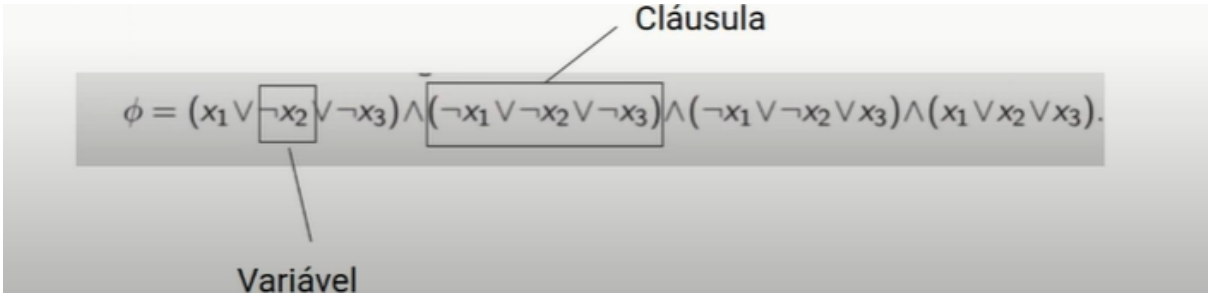
Sabendo que há uma solução para o problema de Cobertura de Vértices, haverá uma função sobrejetora F(s) que irá transformar o problema Scv em solução de Cobertura Independente Sci em tempo polinomial.

Por fim, a função F(s) irá localizar os vértices ou a solução do Conjunto Independente da seguinte maneira: K = | v | - K', onde K seriam os vértices do Conjunto Independente e K' são os vértices da Cobertura de Vértices, logo, a subtração da quantidade de vértices | v | do grafo representado pelo K' da Cobertura de Vértices, nos gerará o número K do Conjunto Independente. Logo, é possível provar que o problema é um NP-Completo

1.2. Problema Subset-Sum (Decisão)

O problema do Subset-Sum funciona da seguinte maneira: dado um conjunto de inteiros, existe um subconjunto não-vazio cuja soma é 0? Por exemplo, dado o conjunto { −7, −3, −2, 5, 8}, a resposta é sim porque o subconjunto { -3, -2, 5} resulta numa soma que dá zero. O problema é NP-completo, mas iremos provar isso utilizando outro problema de satisfação booleana 3SAT.

O problema 3SAT é dada uma forma booleana que contém variáveis e cláusulas:



Existe alguma atribuição a x1, x2 e x3 para dar para essas variáveis tal que essa expressão seja verdadeira? Esse problema é NP-Completo (Corolário 7.42 Sipser) e por meio dele vamos aplicar uma redução ao Subset-SUM.

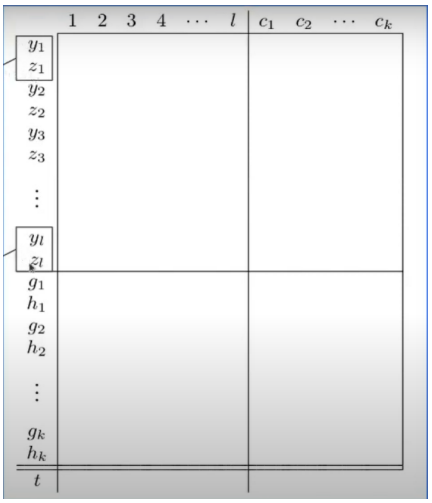
Provando que Subset-SUM está em NP:

Uma forma de provar é dando uma máquina de turing não-determinística de tempo polinomial para Subset-Sum.
N = “Sobre a entrada (S, t):

1. Não-deterministicamente seleciona um subconjunto c dos números em S.
2. Teste se é uma coleção de números que somam t.
3. Se o teste der positivo, aceite; caso contrário, rejeite”

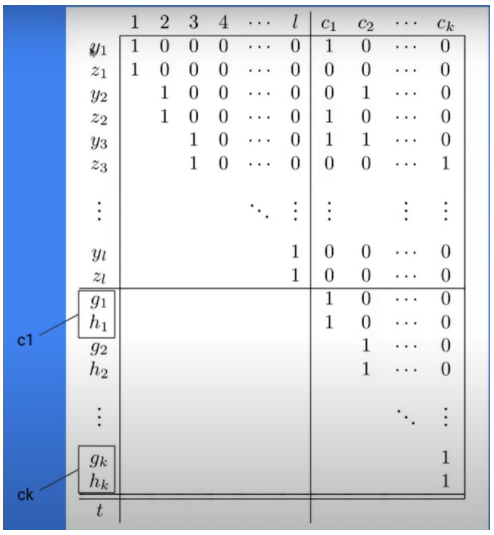
Subset-Sum pode ser reduzido a 3SAT

Seja Φ uma fórmula booleana com as variáveis x1,...,xl e as cláusulas c1,...,ck. A redução converte Φ para uma instância do problema Subset-Sum (S, t) na qual os elementos de S e o número alvo t são as linhas em expressos em notação decimal.

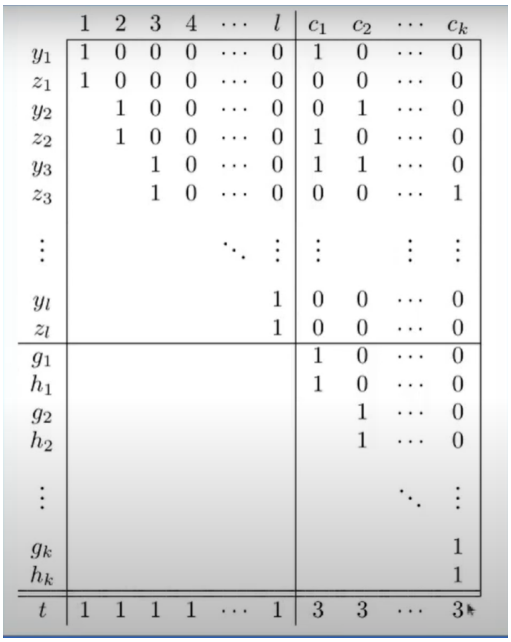


- As linhas acima da linha dupla são os elementos de S.
- A última linha é o número t.
- S contém um par de número yi e zi para cada variável xi em Φ.
- A tabela vai ser preenchida para mostrar as cláusulas de amostra c1, c2 e ck segundo a seguinte expressão booleana:

$$(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3 \vee \cdots) \wedge \cdots \wedge (\overline{x_3} \vee \cdots \vee \cdots).$$



- Adicionalmente, contém pares de números gj, hj para cada cláusula cj.

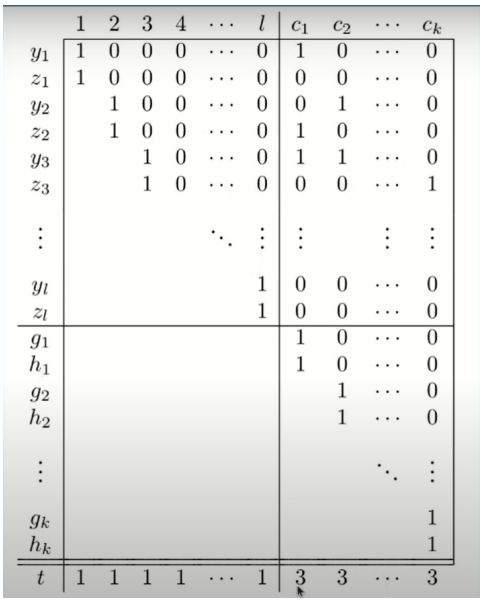


- O número alvo t consiste de l 1s seguidos por k 3s.

Então, para mostrar o motivo que essa construção funciona, podemos demonstrar que Φ é satisfatível se e somente se algum subconjunto de S, ao somar seus elementos, resulta em t.

Vamos supor que Φ seja satisfatível, vamos construir o subconjunto de S que soma em t:

- Então, selecionamos a linha yi caso xi seja V ou a linha zi caso xi seja F na atribuição que satisfaz Φ, o que dá números inteiros.
- Somando esses números que foram selecionados, temos 1 nas primeiras l colunas e um número entre 1 e 3 nas últimas k colunas.
- E a partir disso selecionamos números g e h de tal forma que levem os últimos k dígitos para 3, alcançando o alvo t.
- Agora, vamos supor que um subconjunto de S soma em t, todos os dígitos de elementos de S são 1 ou 0.
- Para que tenha 1 nas primeiras l colunas, o subconjunto deve ter zi ou yi para cada i, mas não ambos, isso ocorre porque não temos nenhum carry na nossa soma.



	1	2	3	4	$\cdots$	l	c_1	c_2	$\cdots$	c_k
y_1	1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
z_1	1	0	0	0	$\cdots$	0	0	0	$\cdots$	0
y_2		1	0	0	$\cdots$	0	0	1	$\cdots$	0
z_2		1	0	0	$\cdots$	0	1	0	$\cdots$	0
y_3			1	0	$\cdots$	0	1	1	$\cdots$	0
z_3			1	0	$\cdots$	0	0	0	$\cdots$	1
$\vdots$					$\ddots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
y_l						1	0	0	$\cdots$	0
z_l						1	0	0	$\cdots$	0
g_1							1	0	$\cdots$	0
h_1							1	0	$\cdots$	0
g_2								1	$\cdots$	0
h_2								1	$\cdots$	0
$\vdots$									$\ddots$	$\vdots$
g_k										1
h_k										1
t	1	1	1	1	$\cdots$	1	3	3	$\cdots$	3

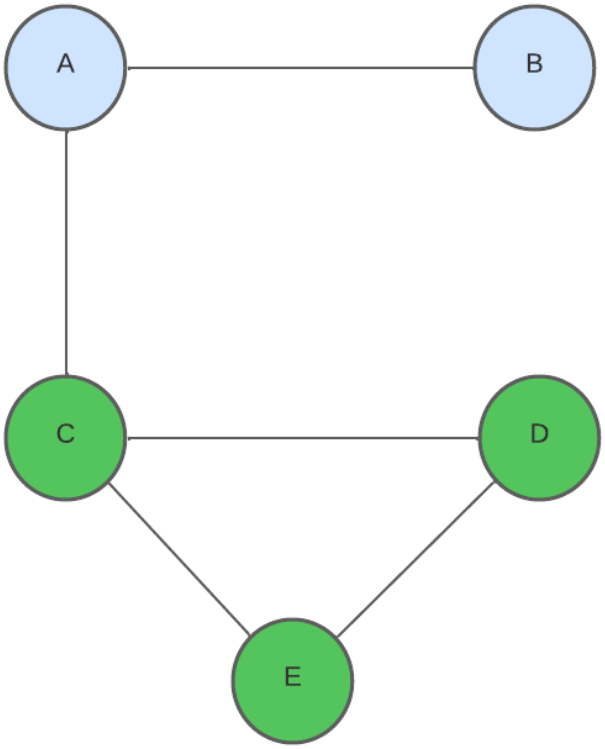
Então concluímos o seguinte:

- Se o subconjunto contém yi, atribuímos V para xi;
- Se o subconjunto contém zi, atribuímos F para xi;
- E isso satisfaz pois a soma nas colunas c1-ck sempre soma em 3 e como as linhas gj e hj somam no máximo 2 nas colunas cj, pelo menos yi ou zi deve ter 1 na coluna cj.
- Se o subconjunto contém yi, então xi possui o valor V e aparece não barrado na cláusula cj;
- Se o subconjunto contém zi, então xi possui o valor F e aparece barrado na cláusula cj;
- De ambas as formas o cj é satisfeito, então Φ é satisfeito.

Finalmente, a redução pode ser feita em tempo polinomial para que o Subset-Sum seja NP-Completo.
A tabela tem o tamanho de (2l+2k+1) linhas e (l+k) colunas, tendo um tamanho total de $2l^2+2k^2+4lk+l+k$, o que se aproxima de $(l+k)^2$. E como cada entrada pode ser facilmente calculada para todo Φ, tempo total é $O(n^2)$ e portanto polinomial e assim o problema Subset-Sum é NP-Completo.

2. Considere o problema CLIQUE restrito a grafos nos quais todo vértice tem grau no máximo 3. Chame este problema de CLIQUE-3. Prove que CLIQUE-3 está em NP.

Considerando o seguinte grafo:



É possível notar que o CLIQUE máximo desse grafo é igual a 3 (representado em verde).

Para que um problema pertença à classe NP, é necessário que o problema em questão tenha uma verificabilidade em tempo polinomial, sendo dado um certificado, é necessário, a partir de uma verificação em tempo polinomial se é uma solução válida.

Certificado - Conside um certificado de um conjunto S que contenha nós no clique e que S (Grafo verde) seja um subgrafo de G (Grafo completo).
Verificação -

Para a verificação, é necessário fazer a confirmação se já um clique de tamanho k (nesse caso, k = 3) no grafo. Logo, verificar se o número de nós em S é igual a k levará um tempo O(1)
Logo após, deve-se verificar se todos os vértices têm um grau de saída igual a k - 1 (no caso, 3 - 1 = 2) levando um tempo $O(k^2)$.
Por fim, para verificar se o grafo formado pelos k (k = 3) nós em S (Grafo verde) está completo, é necessário que seja levado em consideração um tempo $O(k^2) = O(n^2)$.

Concluindo, o problema do CLIQUE-3 pode ser verificado em tempo polinomial, pertencendo assim, à classe NP.