Set Cover – NP-Completo

Eduardo Martins da Rocha

O problema Set Cover: Um Exemplo

$U = \{a, b, c, d, e, f\}$

$S=\{S_1, S_2, S_3, S_4, S_5, S_6\}$

$S = \{\{a, b\}, \{c, f\}, \{b\}, \{d,e\}, \{a,e\}, \{b,f\}\}$

Um subconjunto de S que cobre U

$R=\{\{a,b\}, \{c,f\}, \{d,e\}\}$

O problema de decisão Set Cover

Dado um conjunto U e um conjunto S, formado por subconjuntos de U, é possível selecionar k elementos de S de tal forma que a união destes elementos contenha todos elementos de U?

Demostração de que o problema de decisão Set Cover é NP-Completo

- São necessárias duas provas.
  - Provar que o problema de NP
  - Provar que o problema é NP-Difícil

O problema é NP

- É preciso encontrar um algoritmo que teste uma solução, R, do problema em tempo polinomial
  - Mostrar que R pertence a S
  - Mostrar que R tem K ou menos elementos
  - Mostrar que a união dos elementos de R forma um conjunto igual a U. De outro modo, mostrar que todos os elementos de U estão presentes nos elementos de R.

Elementos de R pertencem a S

```
VerificaSolucao ((U, S , k ), (R))
    contains = true
    for each Ri in R do
        exists = false
        for each Si in S do
            if Si = Ri then
                exists = true
            end
        end
        if exists = false then
            contais = false
        end
    end
    if contains = false then
        return false
    end
```

R contem K ou menos elementos

```
if length (R) > k then
    return false
end
```

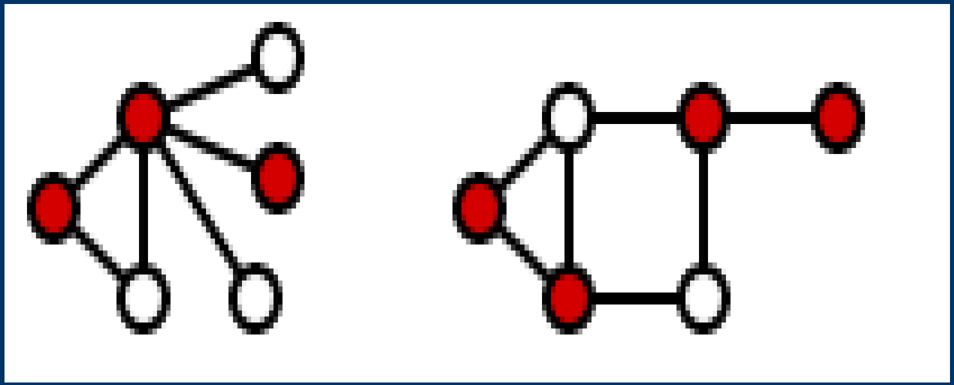
Todos elementos de U estão presentes nos elementos de R

```
        cover = true
        for each Ui in U do
            exists = false
            for each Ri in R do
                for each wi in Ri do
                    if wi = Ui then
                        exists = true
                    end
                end
            end
            if exists = false then
                cover = false
            end
        end
        if cover = false then
            return false
        end
    end
```

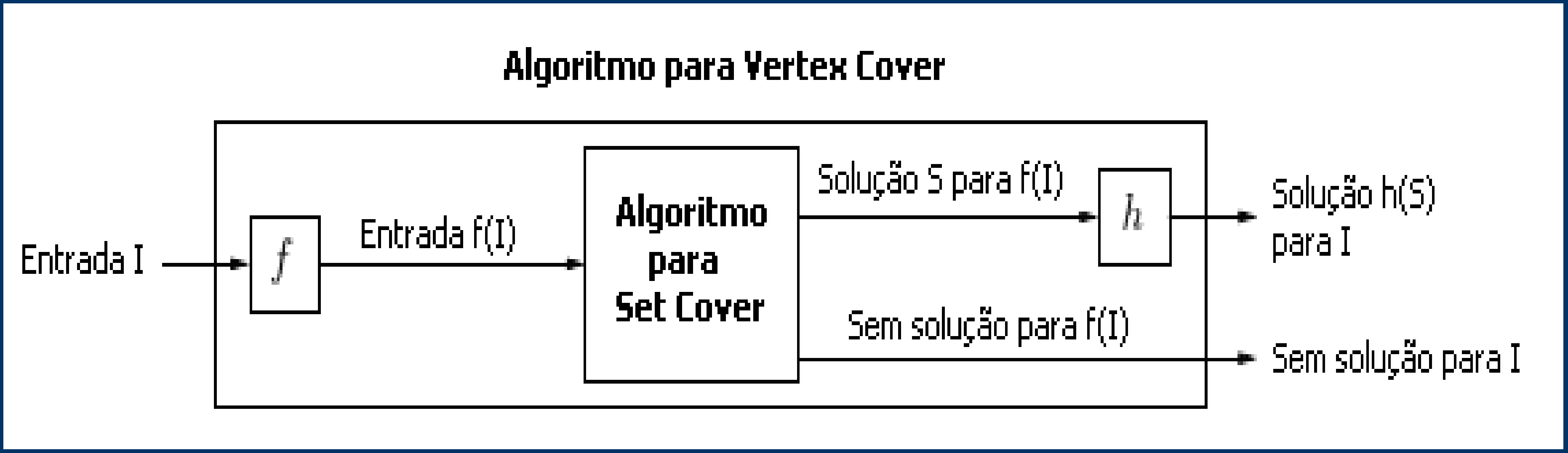
Set Cover é NP-Difícil

- Para provar que o problema de decisão Set Cover é NP-Difícil, o problema de decisão Vertex Cover, um problema NP-Completo, será reduzido ao problema Set Cover.

Problema Vertex Cover



Esquema da Redução

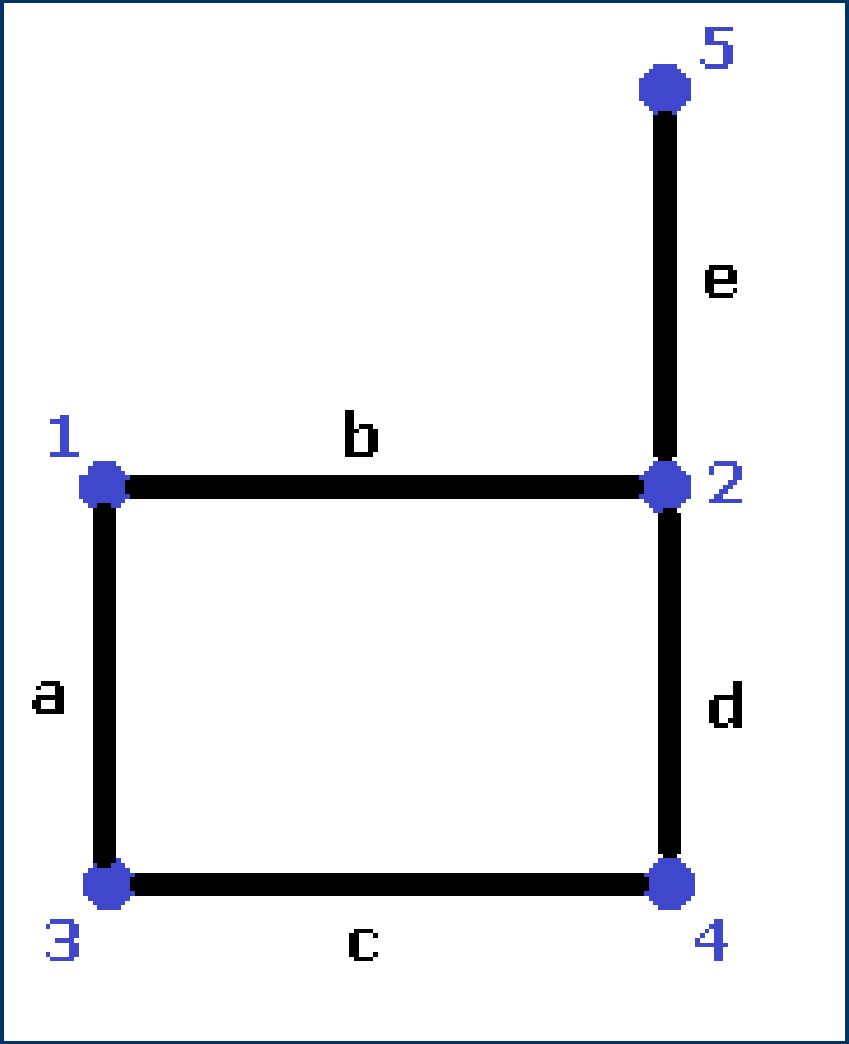


Provar que a redução funciona

- Provar que se
  - Vertex Cover = verdadeiro => Set Cover = verdadeiro
  - Vertex Cover = falso => Set Cover = falso

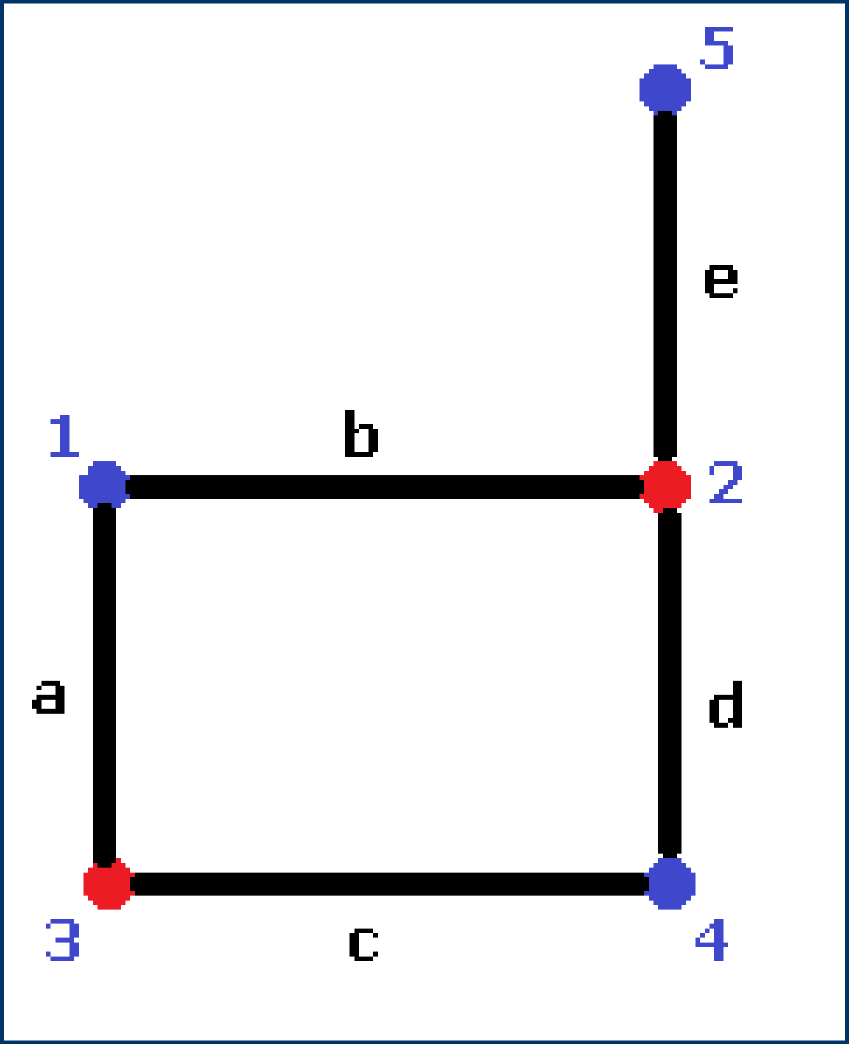
- Ou provar que se
  - Vertex Cover = verdadeiro => Set Cover = verdadeiro
  - Set Cover = verdadeiro => Vertex Cover = verdadeiro

Transformação



- $U = \{a, b, c, d, e\}$
- $S = \{S_1, S_2, S_3, S_4, S_5\}$
- $S_1 = \{a, b\}$
- $S_2 = \{b, d, e\}$
- $S_3 = \{a, c\}$
- $S_4 = \{c, d\}$
- $S_5 = \{e\}$

Transformação



- $W = \{2, 3\}$

- $R = \{S_2, S_3\}$
- $R = \{\{b, d, e\}, \{a, c\}\}$

Suponha que o problema Vertex Cover tenha uma solução W de tamanho K para um grafo $G(V, E)$. Essa solução pode ser transformada em uma solução R do problema Set Cover.

Primeiro, por construção, o $n(R) = K$.

Segundo, os elementos de R cobrem U. Para demostrar isso considere qualquer elemento de U. Esse elemento é uma aresta de G e, como W cobre G, existe pelo menos um vértice em W que é adjacente a esta aresta. Por contrução existe um subconjunto de R que possui este elemento de U.

Agora suponha que exista uma solução R de tamnho K para o problem Set Cover para uma data entrada tranformada de G. Essa solução pode ser transformada em uma solução W para o problema Vertex Cover.

Primeiro, por contrução, $n(W) = K$.

Segundo, W cobre o grafo G. Para demostrar isso considere uma aresta e, qualquer, de G. Sabe que os elementos de U correspondem a todas arestas de G. Sabe também que R cobre U. Ou seja, R deve conter ao menos um subconjunto que possui o elemento e. Por construção, os conjuntos que possuem o elemento e correspondem aos vértices que são adjacentes a e. Por isso, essa aresta e está coberta.

Fim