Problema da Cobertura de Vértices

(Vertex Cover Problem)

Profª:
Dra. Márcia Aparecida Fernandes

Alunos:

Ângelo Neto Travizan

Laura Ferreira Marquez

Leonard Vieira Martins

O problema

- Formalmente:

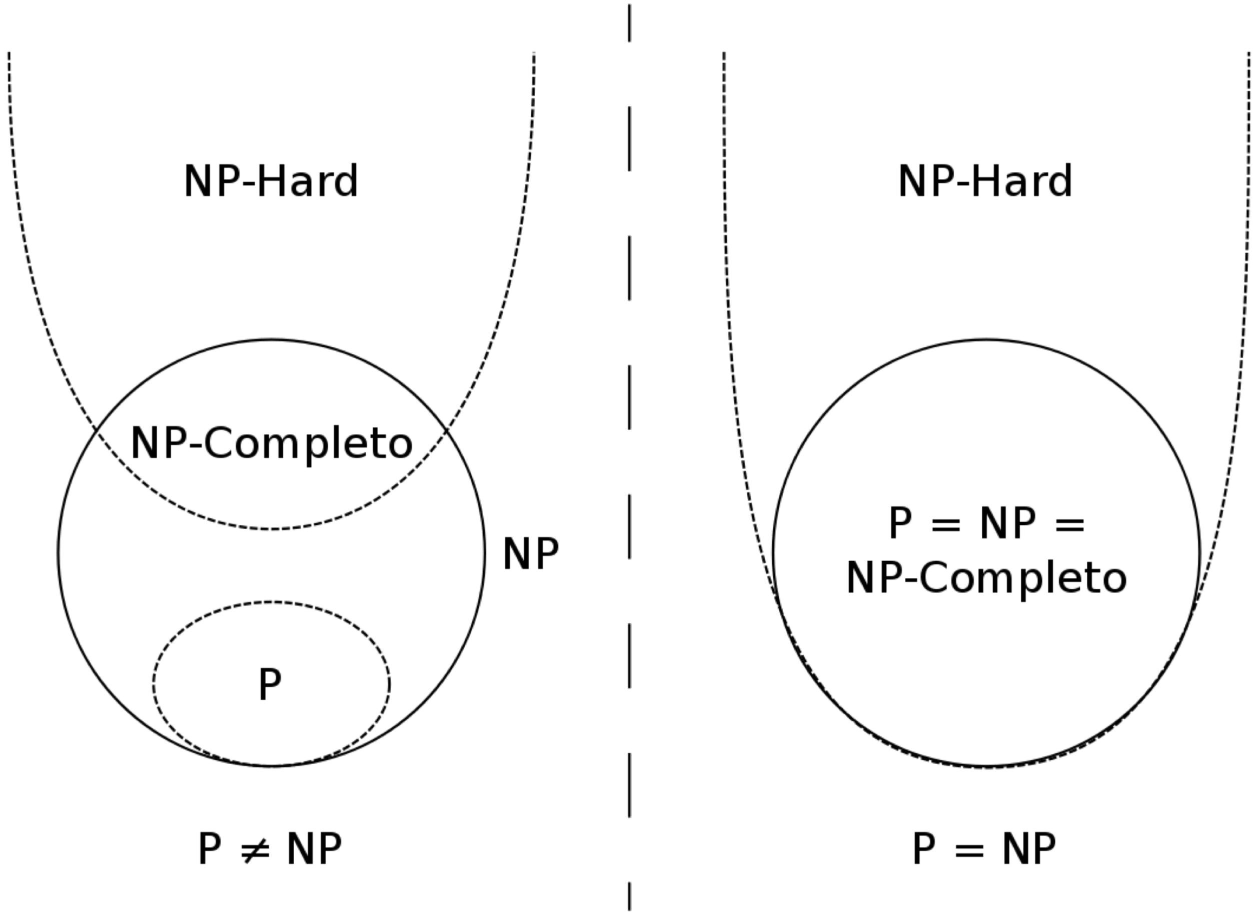
- Seja G=(V, E) um grafo não orientado;

- Uma cobertura de vértices é um subconjunto **V’** contido em **V** tal que, se (**u**, **v**) é uma aresta de **G**, então **u** pertence a **V’** ou **v** pertence a **V’** (ou ambos);

- O tamanho de uma cobertura de vértices é o **número de vértices** que ela contém.

NP-Completo

- Cobertura dos vértices é um problema **NP-Completo**.



NP

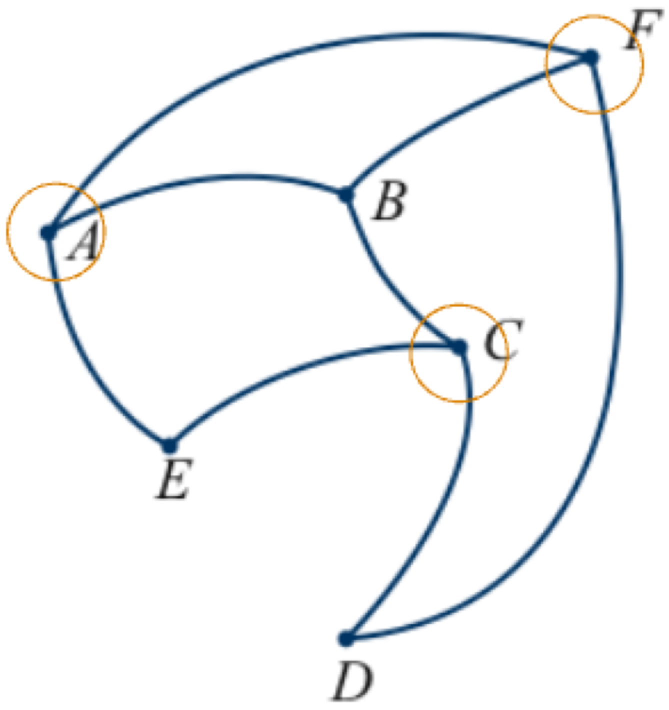
- Classe de complexidade;
- Subconjunto dos problemas de decisão;
- Tempo polinomial não determinístico (Non-Deterministic Polynomial Time).

Simplificado

- Dado um grafo **G=(V, E)**, para todas as arestas de **G**, pelo menos uma das extremidades de cada aresta está na cobertura;
- O ideal seria encontrarmos o **menor** número de vértices possíveis para tal problema…
- Isso significa que a **Solução Ótima** será aquela que contém a **menor** quantidade de vértices que cobrem todas as arestas.

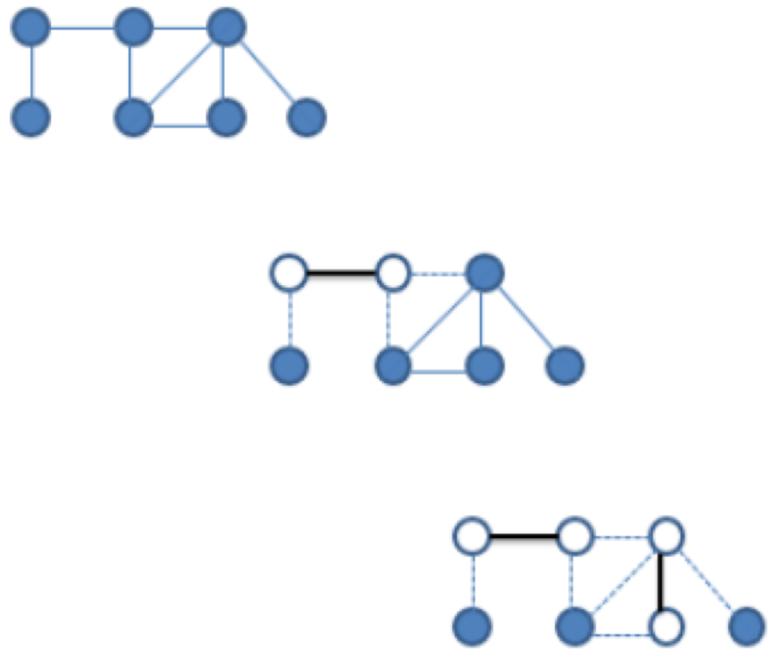
Cobertura de Vértices

Exemplo



Cobertura de Vértices

Exemplo



Por que cobríamos os vértices?

- Aplicações práticas para cobertura de vértices:
- Problemas de localização:
  - Qual melhor lugar para se colocar uma antena ou câmera de segurança;
  - Qual o menor número de guardas deve-se contratar para vigiarem um determinado ambiente;

Algoritmo de Força Bruta

- Algoritmo_coberturaOtima(G):

- 1 - V' = V(G)
- 2 - E' = E(G)
- 3 - para cada v em V':
- 4 - para cada permutação de V', de tamanho 1 a n:
- 5 - E'' = E'
- 6 - para cada vértice da permutação:
- 7 - E'' = E'' - (u, X)
- 8 - para toda aresta incidente a v:
- 9 - E'' = E'' - (v, X)
- 10 - retorne C

Análise do algoritmo de Força Bruta

- Este algoritmo tem como entrada um Grafo, não direcionado e não ponderado. Cuja a saída é a **solução ótima** do problema de Cobertura de Vértices;

- Para essa Análise, considere **V** o número de vértices, e **E** o número de arestas;

- As linhas **1** e **2** extraem do grafo o conjunto de vértices e de arestas, respectivamente, com custo constante.

1 – V' = V(G)
2 – E' = E(G)

Análise do algoritmo de Força Bruta

- Das linhas **3** a **10**, tem-se a seguinte divisão: a linha 3 mostra que as linhas **4** a **10**, serão executadas V vezes.

- **3** - para cada v em V':
- **4** - para cada permutação de V', de tamanho 1 a n:
- **5** - E'' = E'
- **6** - para cada vértice da permutação:
- **7** - E'' = E'' - (u, X)
- **8** - para toda aresta incidente a v:
- **9** - E'' = E'' - (v, X)
- **10** - retorne C

- Na linha **4**, realiza-se uma operação de permutação, que gera 2^V itens.

Análise do algoritmo de Força Bruta

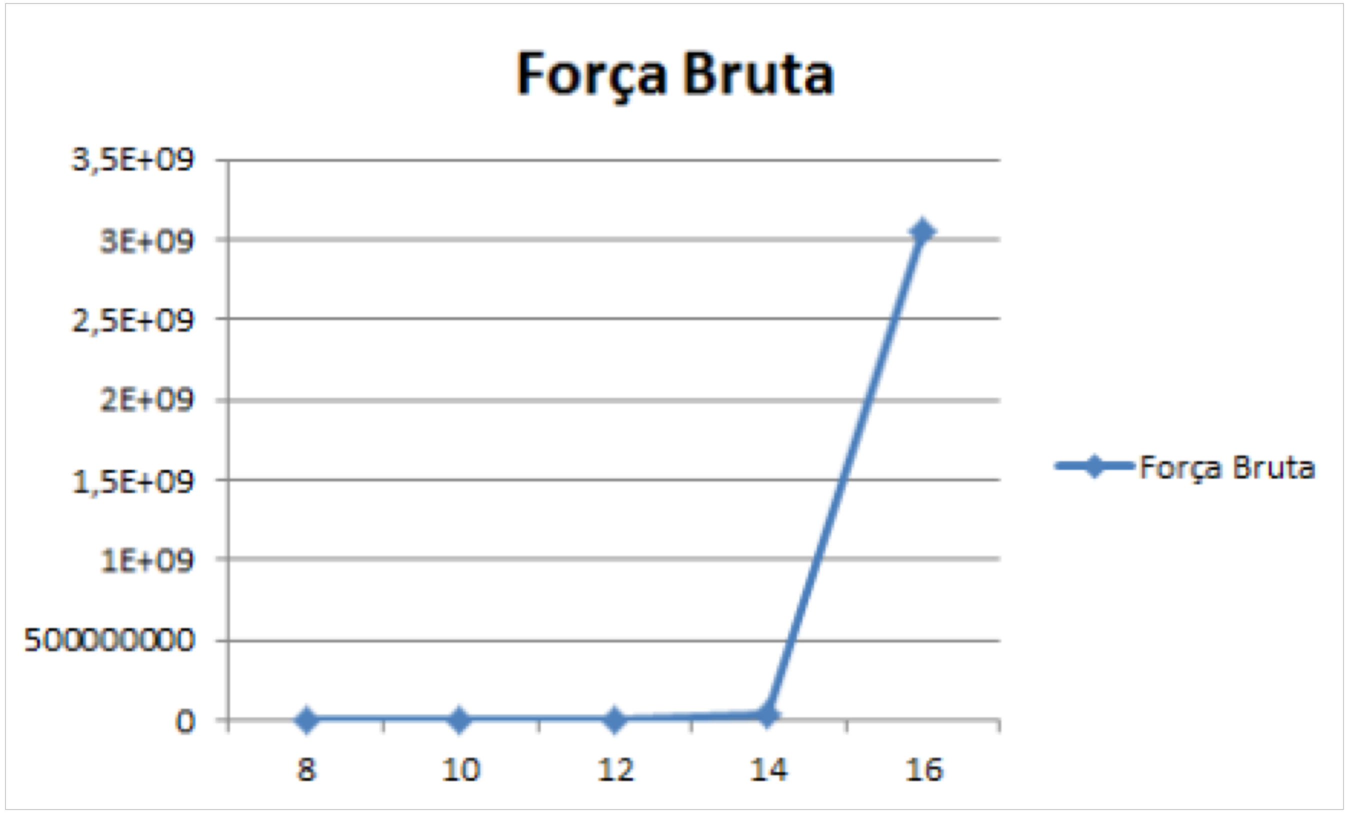
- Das linhas **6** a **9**, basicamente, verifica se cada tupla retornada pela operação de permutação, já realiza ou não a cobertura de todo o grafo. Caso seja verdade, retorna-se essa tupla.

6 - para cada vértice da permutação:
7 - E'' = E'' - (u, X)
8 - para toda aresta incidente a v:
9 - E'' = E'' - (v, X)

- Isto é feito em tempo O(V+E);

- Juntando tudo isso, tem-se um tempo O(V * 2^V *(V+E)).

Análise do algoritmo de Força Bruta



Análise do algoritmo de Força Bruta

- Sempre retorna a **solução ótima**, ou seja, solução composta pelo **menor** número de vértices;

- Custo muito **elevado**;

- **Problema de combinação**: Torna-se todos os subconjuntos possíveis de vértices do grafo, começando pelos **menores**, e verifica se aquele subconjunto realiza a cobertura de todas as arestas do grafo. Se realizar, ele é a resposta ótima, pois o algoritmo toma dos **menores** subconjuntos para os **maiores**.

Algoritmo Cobertura_Aproximado (G, V, E)

- Algoritmo_coberturaAproximada(G):

- 1 - C = {}
- 2 - E' = E(G)
- 3 - enquanto E' != {}
- 4 - (u, v) = aresta de E'
- 5 - C = C + {u, v}
- 6 - para toda aresta incidente a u:
- 7 - E' = E' - (u, X)
- 8 - para toda aresta incidente a v:
- 9 - E' = E' - (v, X)
- 10 - retorne C

Análise do Algoritmo Aproximado

- A entrada do algoritmo é um grafo não direcionado e não ponderado, representado utilizando lista de adjacência;

- A saída é o conjunto de vértices da cobertura de vértices por arestas aproximado.

Análise do Algoritmo Aproximado

- A linha **1** do algoritmo inicia a cobertura C com nenhum vértice, com custo constante;
- **1** - C = {}
- A linha **2** extrai do grafo o conjunto de arestas, com custo de percorrer a lista de adjacência uma vez, com custo proporcional a quantidade de arestas na lista O(E) ou a quantidade de vértices, caso o grafo tenha mais vértices do que arestas O(V). Logo, temos O(V+E), para construir este conjunto.

  2 - E' = E(G)

Análise do Algoritmo Aproximado

- A linha **3** é executada para, no máximo, todas as arestas do grafo, logo o custo máximo do laço é O(E);
- **3** - enquanto E' != {}
- A linha **4** seleciona uma aresta qualquer, tendo custo constante;
- **4** - (u, v) = aresta E'
- A linha **5** adiciona os 2 vértices incidentes na aresta selecionada na cobertura C, com custo constante;
- **5** - C = C + {u, v}

Análise do Algoritmo Aproximado

- Das linhas **6** a **9**, removem as arestas incidentes nos vértices **u** e **v**, varrendo a lista **E'**. Tal operação tem custo proporcional a quantidade de arestas O(E);

- **6** - para toda aresta incidente a u:
- **7** - E'= E' - (u, X)
- **8** - para toda aresta incidente a v:
- **9** - E' = E'- (v, X)

- A linha **10** apenas retorna a cobertura construída, tendo um custo constante;

- **10** - retorne C

Análise do Algoritmo Aproximado

- Portanto, o tempo de execução do algoritmo é O(V+E), considerando que o grafo está representado em lista de adjacência, e suas arestas foram extraídas desta.

Razão de Aproximação

- Já foi mostrado que o Algoritmo Aproximado é executado em tempo polinomial;

- Para ver se ele retorna uma cobertura que é no **máximo** duas vezes o tamanho de uma cobertura ótima, seja A, o conjunto de arestas dadas ao algoritmo. Para cobrir as arestas em A, qualquer cobertura, em especial uma cobertura ótima **C***, deve-se incluir pelo **menos** uma extremidade de cada aresta em A.

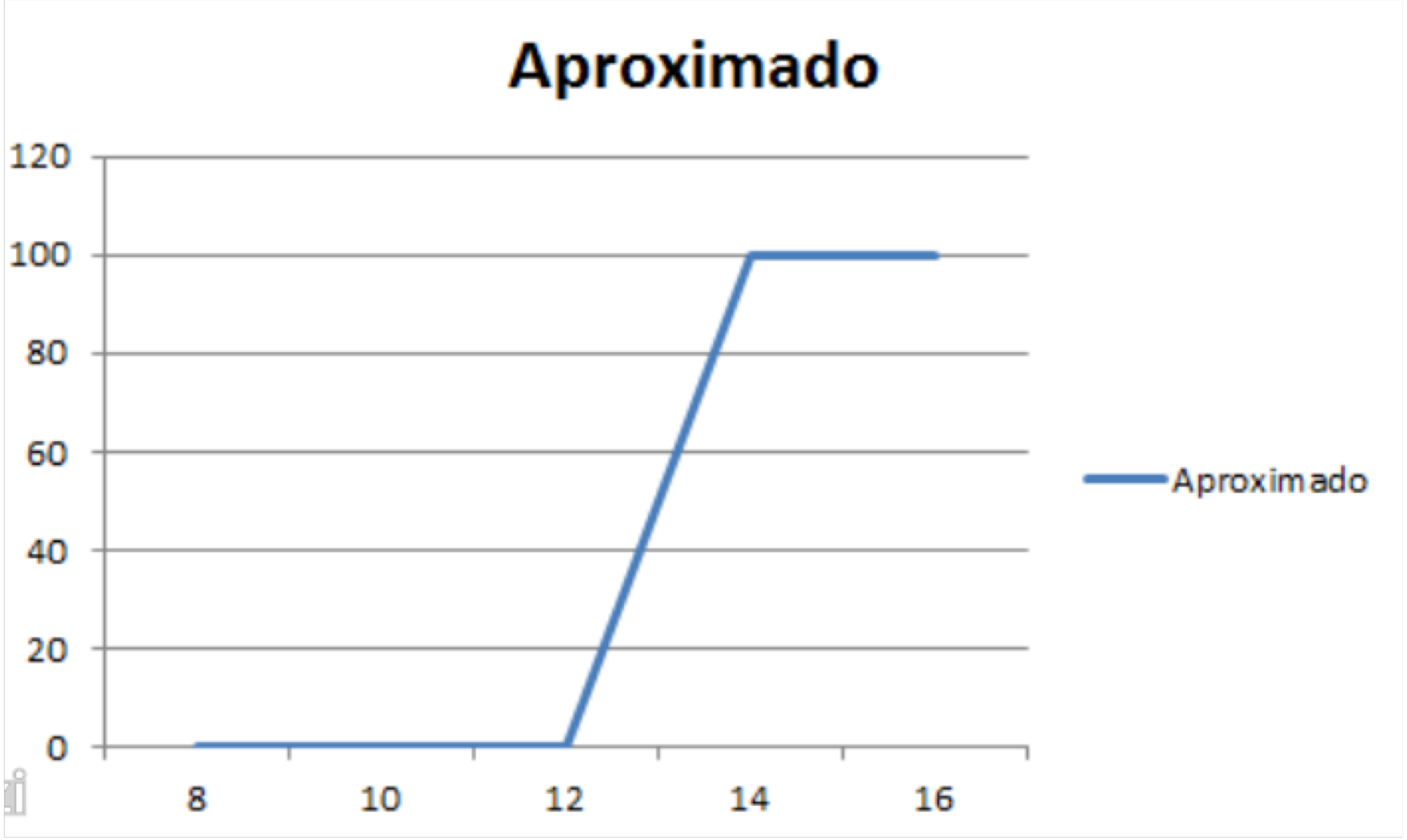
Razão de Aproximação

- Duas arestas em A não compartilham uma extremidade, pois uma vez que uma aresta é escolhida, todas as outras que são incidentes são **eliminadas** pelo algoritmo. Desse modo, não há duas arestas cobertas pelo **mesmo** vértice de C*;
- Encontramos assim, um limite inferior:
- **|C*| >= |A|** sobre o tamanho de uma cobertura ótima.

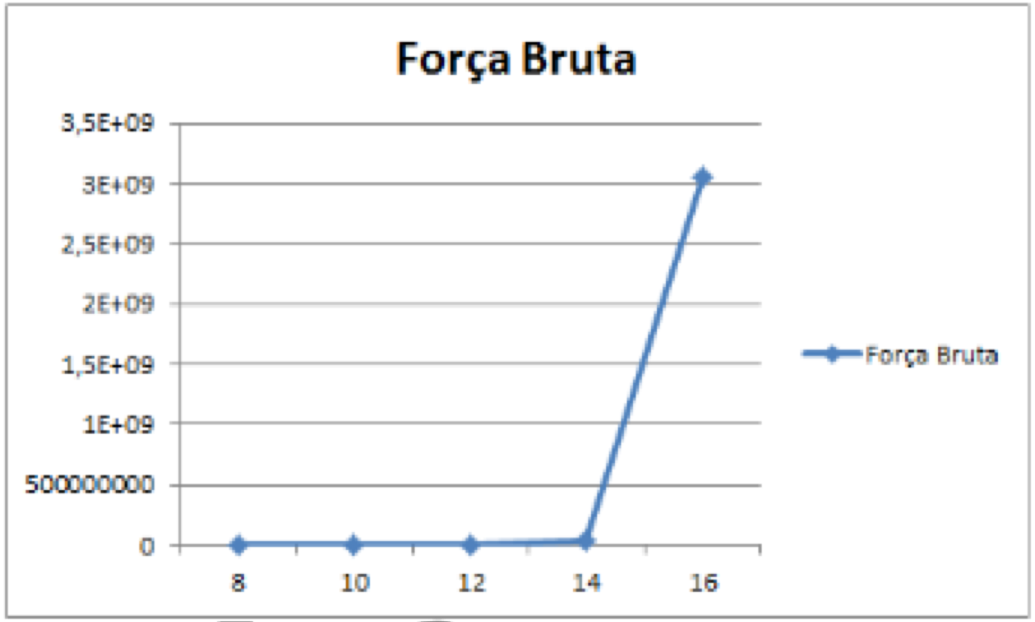
Razão de Aproximação

- Cada iteração do loop escolhe uma aresta que não possui nenhuma das extremidades em C, assim, encontramos um **limite superior** sobre o tamanho da cobertura de vértices retornada:
- **$|C| = 2\ |A|$**
- Combinado as equações, temos:
- **$|C| = 2\ |A| <= 2\ |C^*|$**
- Provando assim, a **razão de aproximação 2**.

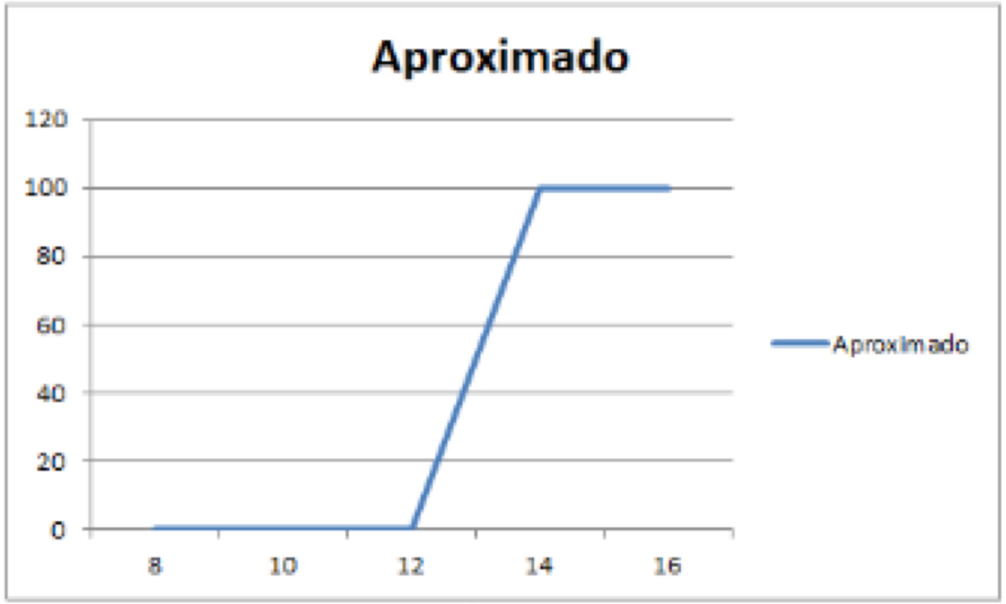
Razão de Aproximação



Conclusão

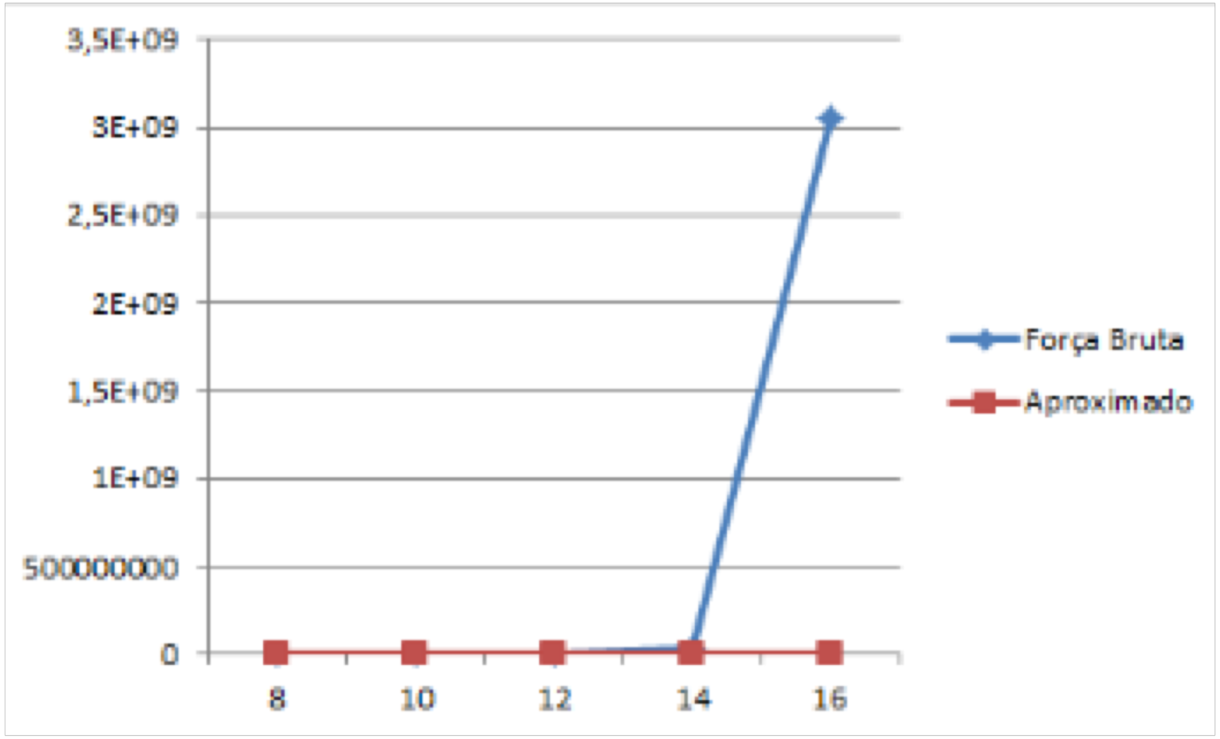


Força Bruta



Aproximado

Conclusão


3,5E+09
3E+09
2,5E+09
2E+09
1,5E+09
1E+09
500000000
0
8
10
12
14
16
Força Bruta
Aproximado

Referências Bibliográficas

- http://pt.wikipedia.org/wiki/Cobertura_de_v%C3%A9rtices_(teoria_dos_grafos)

- **Algoritmos: teoria e prática / Thomas H. Cormen**. Tradução da segunda edição [americana] Vandenberg D. deSouza – Rio de Janeiro : Campus, 2002

- http://www.cs.dartmouth.edu/~ac/Teach/CS105-Winter05/Notes/wan-ba-scribe.pdf

- Notas de aula de Análise de Algoritmo

Cobertura de Vértices

- Obrigado! :)
- Dúvidas?!