



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

MARIANA OLIVEIRA DA SILVA

**PROBLEMA DE COBERTURA POR VÉRTICES EM REDES  
COMPLEXAS**

DISSERTAÇÃO DE MESTRADO

CURITIBA

2013

MARIANA OLIVEIRA DA SILVA

**PROBLEMA DE COBERTURA POR VÉRTICES EM REDES  
COMPLEXAS**

Dissertação apresentada ao Programa de Pós-graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Computação Aplicada” – Área de Concentração: Computação Aplicada.

Orientador: Prof. Dr. Murilo Vicente Gonçalves da Silva

Co-orientador: Prof. Dr. Gustavo Alberto Giménez Lugo

**CURITIBA**

**2013**

Dedico todo meu trabalho à minha mãe, Castorina Oliveira.

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus, por ter colocado esta maravilhosa oportunidade em meu caminho e ter me dado forças para levar este desafio até o fim.

Dedico este trabalho aos meus pais e meu irmão, que sempre me apoiaram em todos os momentos, sempre me deram uma palavra de conforto e carinho nas horas mais conflitantes da minha vida e muitas vezes sem mesmo saberem. E sempre fizeram tudo para me dar uma boa educação e para que nada me faltasse me ajudando a alcançar mais uma meta.

Ao meu noivo, por todo amor, carinho e compreensão que recebi ao longo desse trabalho, que sempre quis me ajudar em todos os momentos da melhor maneira possível.

Ao Professor Dr. Murilo V. G. da Silva (orientador) e ao professor Dr. Gustavo Alberto Giménez-Lugo (co-orientador), pela paciência, dedicação, ajuda e compreensão em tudo que precisei nesse trabalho e também pela confiança depositada em mim.

Aos meus grandes amigos, por não me abandonarem mesmo quando estive ausente por tanto tempo, e a todos aqueles que direta ou indiretamente tenham contribuído para a execução dessa dissertação de mestrado.

## RESUMO

A teoria dos grafos é uma ferramenta matemática muito utilizada na resolução de problemas algorítmicos e computacionais em que se quer modelar conjuntos de elementos e relações entre estes elementos. Sistemas naturais e tecnológicos de diversos domínios podem ser representados matematicamente por grafos que possuem propriedades estatísticas bem conhecidas, sendo uma destas propriedades a distribuição de graus dos vértices do grafo seguindo a lei de potência (power law). Exemplos destes grafos, conhecidos como *grafos power law* são a internet, World-Wide Web, as redes sociais, redes biológicas. No contexto de problemas algorítmicos em grafos, estamos interessados em problemas computacionalmente difíceis de serem resolvidos que pertencem à classe NP-Difícil (ou NP-Hard), mais especificamente no problema de cobertura por vértices. Neste trabalho será estudado experimentalmente o comportamento de um algoritmo baseado em uma estratégia gulosa para o problema de cobertura de vértices e compararemos com outro algoritmo de aproximação e com a solução exponencial ótima. Em particular esta solução será aplicada e analisada em redes complexas.

**Palavras-chave:** Grafos, Redes Complexas, Complexidade Computacional, Cobertura de Vértices, Problemas NP-Completos, Algoritmos.

## ABSTRACT

Graph theory is a mathematical tool used in solving many algorithmic and computational problems in that both sets of model elements and relationships between these elements. Most natural and technological systems can be mathematically modeled by graph having many well known properties, in particular the power law distribution of the vertex degree sequence. Examples of such graphs, called *power law graphs* are the Internet, World-Wide Web, social networks, biological networks. In the context of algorithmic problems on graphs, we are interested in problems in class NP-Hard, more specifically in the vertex cover problem. This work will be studied experimentally the behavior of an algorithm based on a greedy strategy for the vertex cover problem and compare with other approximation algorithms and with the exponential optimal solution. In particular this solution will be applied and analyzed in complex networks.

**Keywords:** Graphs, Complex Networks, Computational Complexity, Vertex Cover, NP-Complete Problems, Algorithms.

## LISTA DE FIGURAS

FIGURA 1 – GRAFO G COM 7 VÉRTICES E 8 ARESTAS. ....	14
FIGURA 2 – GRAFO COMPLETO. ....	15
FIGURA 3 – GRAFO H - SUBGRAFO DE G. ....	15
FIGURA 4 – CLASSES P, NP, NP-COMPLETO E NP-DIFÍCIL ....	19
FIGURA 5 – COBERTURA POR VÉRTICES DE UM GRAFO. ....	20
FIGURA 6 – COBERTURA MÍNIMA POR VÉRTICES DE UM GRAFO. ....	20
FIGURA 7 – REDE POWER LAW ....	22
FIGURA 8 – DISTRIBUIÇÃO DE GRAUS. O EIXO X REPRESENTA O VALOR DO GRAU (DE 0 ATÉ O GRAU MÁXIMO) E O EIXO Y REPRESENTA O NÚMERO DE VÉRTICES COM ESTE GRAU. ....	23
FIGURA 9 – PROCESSO DE NASCIMENTO DE UMA REDE POWER LAW: COMEÇANDO COM UM PAR DE VÉRTICES ADJACENTES E EM QUE CADA NOVO VÉRTICE “ESCOLHE” DOIS VÉRTICES DA REDE PARA SE CONECTAR. ....	25
FIGURA 10 – GRAFO BARABÁSI-ALBERT SINTÉTICO COM 10 VÉRTICES E 30 ARESTAS. ....	27
FIGURA 11 – RESULTADO DA COBERTURA - ALGORITMO DE 2 APROX. ....	29
FIGURA 12– EXECUÇÃO DO ALGORITMO DE 2-APROXIMAÇÃO. NO PRIMEIRO PASSO, A ARESTA $E_1\{V_1, V_2\}$ É ESCOLHIDA. TODAS AS DEMAIS ARESTAS INCIDENTES AOS VÉRTICES $V_1$ E $V_2$ $\{E_1, E_2, E_3, E_4, E_6\}$ SÃO COBERTAS. NO SEGUNDO PASSO, É ESCOLHIDA A ARESTA $E_5\{V_3, V_4\}$ , AS ARESTAS INCIDENTES AOS VÉRTICES $V_3$ E $V_4$ $\{E_5, E_7\}$ SÃO COBERTAS. OS VÉRTICES $V_1, V_2, V_3, V_4$ REPRESENTAM A COBERTURA POR VÉRTICES DO GRAFO ....	29
FIGURA 13 – RESULTADO DA COBERTURA - ALGORITMO GULOSO. ....	31
FIGURA 14– EXECUÇÃO DO ALGORITMO GULOSO. NO PRIMEIRO PASSO, O VÉRTICE $V_2$ É ESCOLHIDO. AS ARESTAS INCIDENTES A ELA $E_1, E_3, E_4, E_6$ SÃO COBERTAS. NO SEGUNDO PASSO, É ESCOLHIDA O VÉRTICE $V_3$ , AS ARESTAS $E_5$ E $E_2$ SÃO COBERTAS. POR FIM, NO TERCEIRO PASSO O VÉRTICE $V_5$ COBRE A ÚLTIMA ARESTA RESTANTE: $E_7$ . OS VÉRTICES $V_2, V_3, V_5$ REPRESENTAM A COBERTURA POR VÉRTICES DO GRAFO ....	33
FIGURA 15– GRÁFICO COM OS RESULTADOS DOS 15 PRIMEIROS GRAFOS ANALISADOS. ....	44

## LISTA DE TABELAS

TABELA 1	–	COMPARAÇÃO DE TEMPO DE EXECUÇÃO ENTRE FUNÇÕES POLINOMIAIS E EXPONENCIAIS. ....	17
TABELA 2	–	EXEMPLOS DE REDES DE POWER LAW. ....	24
TABELA 3	–	GRAU DOS VÉRTICES - PRIMEIRA ITERAÇÃO. ....	32
TABELA 4	–	GRAU DOS VÉRTICES - SEGUNDA ITERAÇÃO. ....	32
TABELA 5	–	GRAU DOS VÉRTICES - TERCEIRA ITERAÇÃO. ....	32
TABELA 6	–	REDES REAIS UTILIZADAS NOS EXPERIMENTOS. ....	41
TABELA 7	–	RESULTADOS DOS EXPERIMENTOS. ....	42
TABELA 8	–	RESULTADOS DOS EXPERIMENTOS ....	43
TABELA 9	–	TRABALHOS FUTUROS. ....	46

## **LISTA DE SIGLAS**

NP      Non Deterministic Polynomial Time

P      Deterministic Polynomial time

## LISTA DE SÍMBOLOS

$\Theta$	Theta
$\varepsilon$	Epsilon
$\lambda$	Lambda
$\subseteq$	Subconjunto
$d(v)$	Grau do vértice $v$
$\Delta(G)$	Grau Máximo de um Grafo $G$
$\delta(G)$	Grau Mínimo de um Grafo $G$
$\in$	Pertence a

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	GRAFOS POWER LAW E OTIMIZAÇÃO COMBINATÓRIA	13
<b>2</b>	<b>GRAFOS, REDES COMPLEXAS E COMPLEXIDADE COMPUTACIONAL</b>	<b>14</b>
2.1	INTRODUÇÃO À TEORIA DOS GRAFOS	14
2.1.1	Representação de Grafos	15
2.2	COMPLEXIDADE: TEMPO POLINOMIAL VS. EXPONENCIAL	16
2.3	PROBLEMAS EM GRAFOS	18
2.4	PROBLEMA DE COBERTURA POR VÉRTICES	19
2.5	REDES COMPLEXAS	20
2.5.1	Leis de Potência ou Power-Law	21
<b>3</b>	<b>COBERTURA POR VÉRTICES EM REDES COMPLEXAS</b>	<b>26</b>
3.1	EXECUÇÃO PASSO A PASSO DE ALGORITMOS PARA COBERTURA POR VÉRTICES	27
3.1.1	Simulação com algoritmo de 2-Aproximação	28
3.1.2	Simulação com Algoritmo Guloso	30
3.1.3	Cobertura de Vértices utilizando o Sage	33
<b>4</b>	<b>RESULTADOS EXPERIMENTAIS</b>	<b>35</b>
4.0.4	Cenários das Simulações	35
4.0.5	Airlines	35
4.0.6	US Air	35
4.0.7	Codeminer	35
4.0.8	Cpan authors	36
4.0.9	EuroSis	36
4.0.10	Oclinks	36
4.0.11	YeastS	36
4.0.12	Grafo CA-GrQc	36
4.0.13	p2p-Gnutella08	36
4.0.14	Wiki-Vote1	37
4.0.15	p2p-Gnutella09	37
4.0.16	p2p-Gnutella06	37
4.0.17	p2p-Gnutella05	37
4.0.18	CA-HepTh	37
4.0.19	p2p-Gnutella04	37
4.0.20	CA-AstroPh	38
4.0.21	p2p-Gnutella25	38
4.0.22	CA-CondMat	38
4.0.23	p2p-Gnutella24	38
4.0.24	Cit-HepTh	38
4.0.25	p2p-Gnutella30	38
4.0.26	Email-Enron	39
4.0.27	Brightkite-edges	39

4.0.28p2p-Gnutella31 .....	39
4.0.29soc-Epinions1 .....	39
4.1 RESULTADOS .....	39
<b>5 CONCLUSÃO .....</b>	<b>45</b>
<b>REFERÊNCIAS .....</b>	<b>47</b>

## 1 INTRODUÇÃO

A teoria de grafos é uma ferramenta matemática simples e poderosa, usada para construção de modelos e resolução de problemas [47], tais como: fluxos em redes, escolha de uma rota ótima entre dois pontos em um mapa, logística [13], representação de imagens [26] e diversos outros. Historicamente o nascimento da área é atribuído ao problema das pontes de Königsberg, solucionado por Leonhard Euler em 1736 [54]. Entretanto foi apenas na segunda metade do século XX, com o advento do computador eletrônico que a teoria dos grafos ganhou grande impulso e desde então é uma das áreas mais ativas de pesquisa em matemática e em fundamentos matemáticos em computação.

O estudo de grandes grafos reais, também chamado de redes complexas, cresceu na última década, reunindo físicos, matemáticos, cientistas da computação e muitos outros pesquisadores [38]. Nesta área, muitos trabalhos experimentais foram realizados por estudiosos [9, 39, 20, 33, 28, 18].

Um ponto crucial neste campo é que muitas redes complexas de diferentes domínios, compartilham propriedades comuns. Algumas das propriedades comuns mais notáveis são: o pequeno diâmetro (o fenômeno do mundo pequeno), a distribuição *power law*, alto coeficiente de agrupamento, entre outros [5, 7].

Redes Complexas podem modelar vários sistemas naturais ou artificiais, como redes sociais, biológicas e tecnológicas, e podem ser representadas matematicamente por grafos com propriedades estatísticas específicas conhecido como grafos (ou redes) *power law*. A distribuição de conectividade destas redes segue leis de potência, o que implica numa minoria dos vértices com muitas ligações e uma abundância de vértices com poucas ligações [2].

No contexto da computação, o foco deste estudo baseia-se em problemas algorítmicos em grafos. Vários problemas clássicos como o problema de caminho mínimo, fluxo máximo, busca e emparelhamento admitem algoritmos eficientes e são amplamente utilizados na prática. Para outros problemas de otimização, apesar de terem grande interesse prático, não se conhecem algoritmos eficientes na prática. Tais problemas, na linguagem da teoria de complexidade com-

putacional são conhecidos como problemas NP-Completo e a comunidade científica acredita que não admitem soluções eficientes [21].

Em 1972, Karp publicou o artigo clássico na área chamado *Reducibility Among Combinatorial Problems*, (Em português: Redutibilidade entre problemas combinatórios) [35], onde descreve 21 problemas NP-Completo [35], sendo que vários deles envolvem grafos. Dentre estes problemas, o problema de cobertura por vértices em sua versão de otimização (NP-Difícil) será o objeto de estudo desta dissertação.

Uma cobertura mínima de vértices é o menor conjunto de vértices que “toca” todas as arestas de um grafo. Neste trabalho, serão realizados experimentos em uma série de grafos reais a partir de bases de dados de redes complexas [23, 43, 49] a fim de encontrar uma “boa” cobertura por vértices (lembrando que encontrar uma cobertura ótima é um problema NP-Difícil). Em particular, explorou-se a distribuição de graus seguindo a *power law* apresentada em muitas redes complexas. Nos experimentos realizados, verificou-se que a execução de um algoritmo guloso em uma rede *power law* pode resultar em uma cobertura por vértices de cerca de 1,02 vezes a cobertura ótima teórica. Este é um resultado prático interessante, pois, teoricamente, sabemos que:

1. Em um grafo qualquer com  $n$  vértices, uma abordagem gulosa não pode garantir a fator melhor do que  $\ln n$ ;
2. O melhor algoritmo de aproximação conhecido até o momento tem um fator muito maior de  $2 - \Theta\left(\frac{1}{\sqrt{\log n}}\right)$ [34]. De fato, no contexto da aproximação com um fator constante, é conjecturado que não existe uma  $(2 - \epsilon)$  aproximação para o problema, onde  $\epsilon$  representa uma constante tão pequena quanto se queira;
3. Mesmo restrito aos grafos *power law* e garantias probabilísticas, a taxa de aproximação mais conhecida é 1.5, ou seja, o resultado da cobertura terá garantia de ser no máximo 1.5 vezes o ótimo [29].

Nesta pesquisa, de modo geral, procura-se responder a seguinte pergunta: É possível explorar estas propriedades comuns à maioria das redes do mundo real, a fim de obter algoritmos mais eficientes para problemas de otimização combinatória? Esta é uma questão prática interessante, que ainda precisa ser explorada mais profundamente. O ponto principal deste estudo é que, uma vez que grafos *power law* contêm poucos “hubs”, ou seja, vértices de grau muito elevado, naturalmente, pode-se esperar que uma abordagem gulosa (escolhendo primeiro estes hubs para a cobertura) terá uma boa performance. Embora esta idéia seja obviamente bastante

natural, não foi encontrado qualquer trabalho experimental ou analítico para confirmar esta expectativa. Mais importante ainda, uma questão que também não vem sendo considerada é como (quantitativamente) será a performance de um algoritmo guloso em grafos reais *Power Law*. Tais experimentos foram executados em 25 grafos de diferentes domínios de bases de dados de redes complexas e foram obtidas coberturas muito próximas do “ótimo”, mais precisamente cerca de 1,02 vezes o ótimo (o maior valor obtido foi de 1,05). São apresentados este e outros resultados relacionados no Capítulo 4.

## 1.1 GRAFOS POWER LAW E OTIMIZAÇÃO COMBINATÓRIA

Barabasi e Albert [5] e também outros [11] observaram que em muitas redes reais, a distribuição de graus dos vértices é baseada em leis de potência, isto é, a fração de vértices com um grau  $d$  é proporcional a  $d^{-\lambda}$ , onde  $\lambda$  é uma constante independente do tamanho do grafo. Trabalhos experimentais nesta área apontaram que o  $\lambda$  mencionado acima é de  $2.9 \pm 0.1$  e Bollobás e outros [8], em 2001, apresentaram um argumento analítico apontando que esse parâmetro é de fato 3. Neste estudo, será utilizada a designação comum “grafo power law” para grafos com esta distribuição de graus.

Muitos grafos reais são de fato grafos power law e pesquisadores da área têm explorado esta propriedade estrutural [45, 22, 40]. Recentemente, alguns trabalhos também abordaram estudos com grafos power law com o objetivo de diminuir o fator de aproximação para problemas NP-Difícil [29, 30, 31]. Mais precisamente, no caso do problema de cobertura por vértices, embora acredite-se que o menor fator constante para um algoritmo de aproximação seja 2, em casos especiais, como grafos power law, Gast e Hauptmann propõem um algoritmo que gera uma cobertura por vértices esperada não maior do que 1.5 vezes a ótima [29]. Estes resultados analíticos estão alinhados com a proposta deste estudo experimental. Em particular, mostramos que, na prática algoritmos simples podem alcançar um fator de aproximação muito menor.

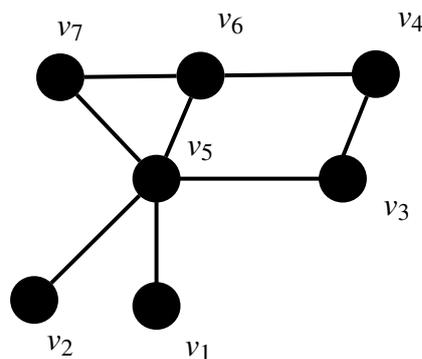
Este trabalho está organizado da seguinte forma: no Capítulo 2 são apresentados conceitos relacionados ao problema de cobertura por vértices, algoritmos gulosos e algoritmos em redes complexas. O Capítulo 2 inclui também a definição de teoria dos grafos, teoria da complexidade, redes complexas, redes power law e o problema de cobertura por vértices em suas duas versões: otimização e decisão. Os detalhes sobre a implementação do Algoritmo Guloso e do Algoritmo de 2-Aproximação, e discussão dos resultados experimentais é realizada no Capítulo 3. Os resultados experimentais são expostos no Capítulo 4. Por fim, o capítulo 5 apresenta as conclusões finais, e inclui as contribuições, limitações e propostas para trabalhos futuros.

## 2 GRAFOS, REDES COMPLEXAS E COMPLEXIDADE COMPUTACIONAL

Este capítulo busca apresentar a fundamentação teórica que será utilizada para tratar o tema ao longo do trabalho. As definições são parcialmente baseadas em [53].

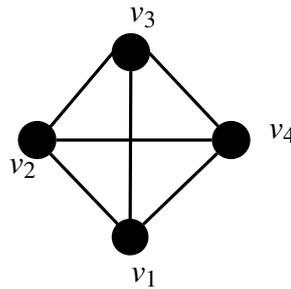
### 2.1 INTRODUÇÃO À TEORIA DOS GRAFOS

Um grafo simples  $G = (V, E)$  consiste de um conjunto finito de vértices  $V(G) = \{v_1, v_2, \dots, v_n\}$  (ou nós) e um outro conjunto  $E(G) = \{e_1, e_2, \dots, e_m\}$ , chamados de arestas, tal que cada aresta de  $E$  é um conjunto de dois vértices de  $V$ . Se  $e = \{v_i, v_j\}$ , dizemos que  $e$  conecta  $v_i$  a  $v_j$ . Demonstramos também a aresta  $v_i, v_j$  da seguinte maneira simplificada:  $v_i v_j$ . Em um grafo  $G = (V, E)$ , se uma aresta  $e$  conecta um vértice  $v_i$  a um vértice  $v_j$ , dizemos que esta aresta é incidente a estes vértices. Na Figura 1 é apresentado um exemplo de grafo com 7 nós e 8 arestas.



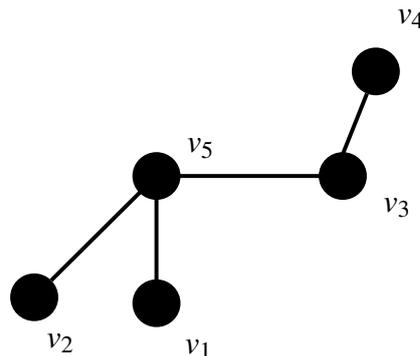
**Figura 1: Grafo G com 7 vértices e 8 arestas.**

Alguns grafos recebem nomes especiais. O grafo  $G$  é denominado vazio quando  $|V(G)| = 0$  e  $G$  é dito trivial se  $|V(G)| = 1$ . Um grafo completo, denotado por  $(K_n)$ , é um grafo não direcionado no qual todos os pares de vértices são adjacentes (Figura 2). Um grafo composto por laços e arestas múltiplas pode ser chamado de multigrafo. Um grafo simples não possui arestas múltiplas e laços. Dado um grafo  $G$ , nos referimos ao seu conjunto de vértices como  $V(G)$  e seu conjunto de arestas como  $E(G)$ .



**Figura 2: Grafo completo.**

No grafo da Figura 1, o conjunto de vértices é  $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$  e o conjunto de arestas é  $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\} = \{v_1v_5, v_2v_5, v_3v_5, v_6v_5, v_6v_4, v_6v_7, v_5v_7, v_3v_4\}$ . Um subgrafo de um grafo  $G$  é um grafo  $H$  cujo conjunto de vértices de  $H$  é um subconjunto de vértices de  $G$ , e o conjunto de arestas de  $H$  é um subconjunto de arestas de  $G$ , ou seja,  $V(H) \subseteq V(G)$  e  $E(H) \subseteq E(G)$ . A Figura 3 apresenta o grafo  $H$  que é um subgrafo do grafo  $G$  apresentado na Figura 1.



**Figura 3: Grafo H - Subgrafo de G.**

O grau de um vértice  $v$  é denotado por  $d_G(v)$  ou  $d(v)$  e refere-se ao número de arestas incidentes ao vértice em questão. O grau máximo de grafo  $G$  é descrito pela notação  $\Delta(G)$ , que é dado por  $\Delta(G) = \max\{d_G(v)\}$ , tal que  $v \in V(G)$ ; e o grau mínimo por  $\delta(G)$ , que é dado por  $\delta(G) = \min\{d_G(v)\}$ , tal que  $v \in V(G)$ . Um grafo  $G$  é dito *regular* se  $\Delta(G) = \delta(G)$ , ou seja, todos os vértices possuem o mesmo grau. Na Figura 3, por exemplo,  $d(v_5) = 3$ ,  $\Delta(G) = 3$  e  $\delta(G) = 1$ .

### 2.1.1 Representação de Grafos

Os grafos podem ser representados de diversas formas, entre elas, as mais utilizadas são: matriz de adjacência, lista de adjacência, lista de incidência e matriz de incidência. Neste estudo, foi utilizada a matriz de adjacência para representar os grafos empregados nos experi-

mentos.

Em relação à matriz de adjacência de um grafo  $G = (V, E)$  com  $n$  vértices: a matriz de adjacência para  $G$  é uma matriz  $A$  bidimensional  $n \times n$ , onde  $A(v_i, v_j) = 1$  se a aresta  $v_i v_j$  está presente em  $G$ .

A matriz  $A$  abaixo representa o grafo  $G$  da Figura 3. Observe que a matriz  $A$  é simétrica. Isso decorre do fato da aresta ser um conjunto de vértices e não um par ordenado.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Além do entendimento dos conceitos básicos relacionado à teoria dos grafos apresentados acima, também é necessário o entendimento dos conceitos em relação à complexidade apresentados na próxima seção.

## 2.2 COMPLEXIDADE: TEMPO POLINOMIAL VS. EXPONENCIAL

Uma função  $f(n)$  é dita função de complexidade de tempo de um algoritmo quando o algoritmo ao tomar como entrada um objeto de tamanho  $n$  ele precisa de  $f(n)$  passos para resolver o problema. Esta função de complexidade de tempo na maioria dos casos tende a ser polinomial ou exponencial [21].

Observa-se na Tabela 1 uma comparação entre tempos que um computador relativamente rápido (capaz de efetuar um milhão de operações aritméticas básicas por segundo) levaria para executar processos com diferentes funções de complexidade.

Os processos das três primeiras linhas executam em tempo polinomial. O tempo de computação aumenta à medida que o conjunto de dados cresce, mas o aumento é razoável e mesmo para um conjunto de dados com tamanho 50, o processo leva apenas uma fração de segundo. Já os processos das duas últimas linhas rodam em tempo exponencial. Neste caso o tempo de computação cresce vertiginosamente.

Para Garey e Johnson [21], a complexidade de tempo da maioria dos problemas é polinomial ou exponencial. Para Devlin, operar em tempo exponencial é extremamente demorado mesmo para os computadores mais rápidos do mundo [16].

- Polinomial: função de complexidade é  $\mathcal{O}(p(n))$ , onde  $p(n)$  é um polinômio. Ex.: algoritmos com pesquisa binária  $\mathcal{O}(\log n)$ , pesquisa sequencial  $\mathcal{O}(n)$ , ordenação por inserção  $\mathcal{O}(n^2)$ , e multiplicação de matrizes (usando um algoritmo ingênuo)  $\mathcal{O}(n^3)$ .
- Exponencial: função de complexidade pode ser representada por  $\mathcal{O}(2^n)$  ou  $\mathcal{O}(3^n)$ , entre outras representações.

**Tabela 1: Comparação de tempo de execução entre funções polinomiais e exponenciais.**

FUNCAO TEMPO/ COM- PLEXIDADE DE $f(n)$	n					
	10	20	30	40	50	60
$n^1$	.00001 seg	.00002 seg	.00003 seg	.00004 seg	.00005 seg	.00006 seg
$n^2$	.0001 seg	.0004 seg	.0009 seg	.0016 seg	.0025 seg	.0036 seg
$n^3$	.001 seg	.008 seg	.027 seg	.064 seg	.125 seg	.216 seg
$2^n$	.001 seg	1.0 seg	17.9 seg	12.7 dias	35.7 anos	366 séculos
$3^n$	.059 seg	58 min	6.5 anos	3855 séculos	$2 \times 10^8$ séculos	$1.3 \times 10^{13}$ séculos

**Fonte: Adaptado de [21]**

Na prática, algoritmos são ditos eficientes se eles podem ser executados em tempo polinomial. Os problemas ditos tratáveis são aqueles que podem ser resolvidos com algoritmos eficientes. Problemas para os quais não se conhecem algoritmos eficientes para resolvê-los são conhecidos como intratáveis [21].

### 2.3 PROBLEMAS EM GRAFOS

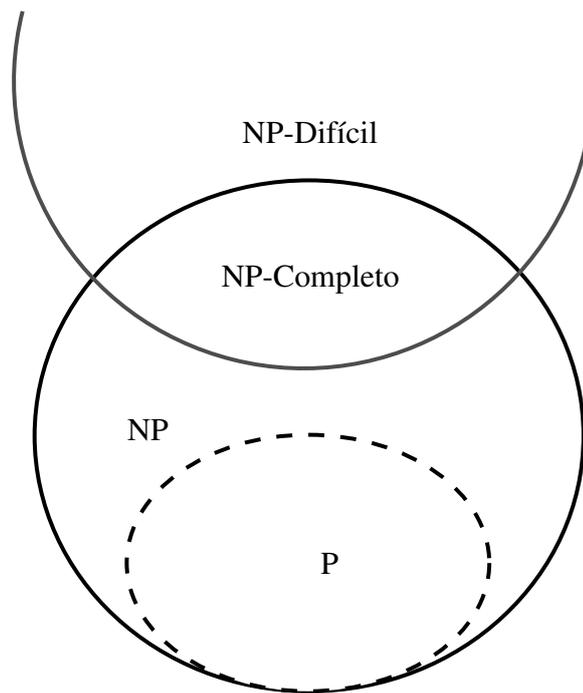
Uma das preocupações centrais da teoria dos grafos é a construção de algoritmos eficientes para a solução de problemas sobre grafos. Muitos desses problemas foram motivados por importantes aplicações práticas.

Existem duas classes notáveis de problemas algorítmicos, denominadas problemas de decisão e problemas de otimização [51]:

- Problemas de decisão: consiste em responder uma questão com Sim ou Não. Exemplo: “existe um caminho entre os vértices  $v_1$  e  $v_2$  de um dado grafo?”.
- Problemas de otimização: consiste em “encontrar estrutura  $S$  que satisfaça critérios de otimização”. Exemplo: “Qual é o menor caminho entre os vértices  $v_1$  e  $v_2$  de um dado grafo?”.

No estudo da Complexidade Computacional os problemas são agrupados em classes, de acordo com sua complexidade. Segundo Garey e Johnson [21], a classe NP é o conjunto de problemas de decisão para os quais, dada uma solução, é fácil (existe algoritmo polinomial) testar se ela é correta. A classe P é o conjunto de problemas onde a busca por uma solução para o problema é feita em tempo polinomial. A classe de complexidade NP-completo é o subconjunto dos problemas de decisão em NP de tal modo que todo problema em NP pode ser reduzido, com uma redução de tempo polinomial, a um dos problemas NP-completo. Pode-se dizer que os problemas de NP-completo são os problemas mais difíceis de NP e que muito provavelmente não admitam solução polinomial e, em particular, caso um destes problemas NP-Completo possa ser resolvido em tempo polinomial, então todos os problemas da classe NP também podem. A comunidade científica conjectura que  $P \neq NP$ . A figura 4 apresenta a distribuição das classes P, NP, NP-Completo e NP-Difícil.

São exemplos de Problemas NP-Completo: Problema de Cobertura por Vértices (dado um grafo  $G$ , e um inteiro  $k$ , retorne se existe uma cobertura - conjunto de vértices que “toca” todas as arestas de um grafo  $G$  - de tamanho  $k$ ); Clique (dado um grafo  $G$  e um  $l$ , retorne se  $G$  contém  $K_l$  como subgrafo); entre outros. No universo dos problemas relacionados aos grafos, um deles foi eleito como objeto deste estudo, que denomina-se Problema de Cobertura Mínima de Vértices, explicado na seção seguinte.



**Figura 4: Classes P, NP, NP-Completo e NP-Difícil**

#### 2.4 PROBLEMA DE COBERTURA POR VÉRTICES

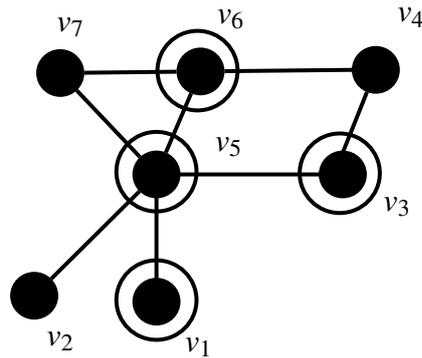
O Problema de Cobertura por Vértices é um problema clássico em ciência da computação e um dos primeiros problemas que foi provado ser NP-Completo [35, 21]. Segundo WEST [53], trata-se de encontrar um conjunto de vértices que “toca” todas as arestas de um grafo, ou seja, dado um grafo  $G$ , pede-se um subconjunto de vértices,  $V' \subseteq V$  tal que para cada aresta  $u, v \in E(G)$ ,  $u \in V'$  ou  $v \in V'$  [50]. O problema da cobertura mínima de vértices refere-se ao menor conjunto de uma cobertura de um grafo.

O problema de Cobertura por Vértices estava entre os 21 problemas no artigo original de Karp [35] que originou boa parte do que hoje se entende por teoria de complexidade computacional.

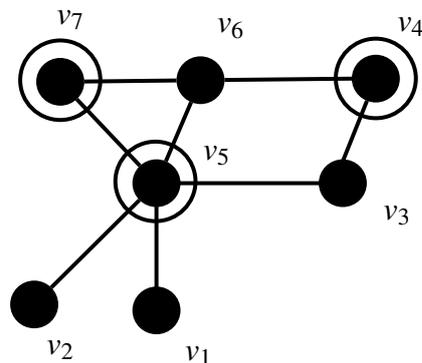
A Figura 5 representa uma das possíveis coberturas de vértices de um grafo. Os vértices  $v_1, v_3, v_5$  e  $v_6$  (vértices em destaque) formam o conjunto de cobertura por vértices:

A Figura 6 representa uma das possíveis soluções de menor cobertura por vértices de um grafo. Os vértices  $v_4, v_5$  e  $v_7$  (vértices em destaque) formam o conjunto de cobertura mínima de vértices.

Existem duas versões para o problema de Cobertura por Vértices: decisão e otimização. A versão de otimização tem como objetivo encontrar a menor cobertura para um grafo dado (Pro-



**Figura 5: Cobertura por Vértices de um grafo.**



**Figura 6: Cobertura Mínima por Vértices de um grafo.**

blema NP-Difícil); enquanto que a versão de decisão tem como objetivo verificar se existe uma cobertura de tamanho específico (Problema NP-Completo). O foco de estudo desta dissertação estará voltado para a versão NP-Difícil do problema de cobertura por vértices.

Após o entendimento do problema foco deste estudo, na próxima seção será realizada uma explanação referente às redes complexas, mais precisamente abordando o tema relacionado as redes power law, nas quais deseja-se realizar uma avaliação de algoritmos para cobertura mínima de vértices.

## 2.5 REDES COMPLEXAS

Durante 40 anos, acreditava-se que a maioria das estruturas da natureza e mesmo os artificiais que têm estrutura de redes (redes biológicas, redes sociais, redes de aeroportos, etc) eram completamente aleatórias. Este paradigma iniciou-se com o trabalho de dois matemáticos húngaros, Paul Erdos e Alfréd Rényi em 1959 [7]. Segundo os estudos desses dois cientistas, em uma rede aleatória, cada par de vértices tem a mesma probabilidade de conexão e esta conexão ocorre de forma independente das demais. Todos os *vértices* têm a mesma chance de receber

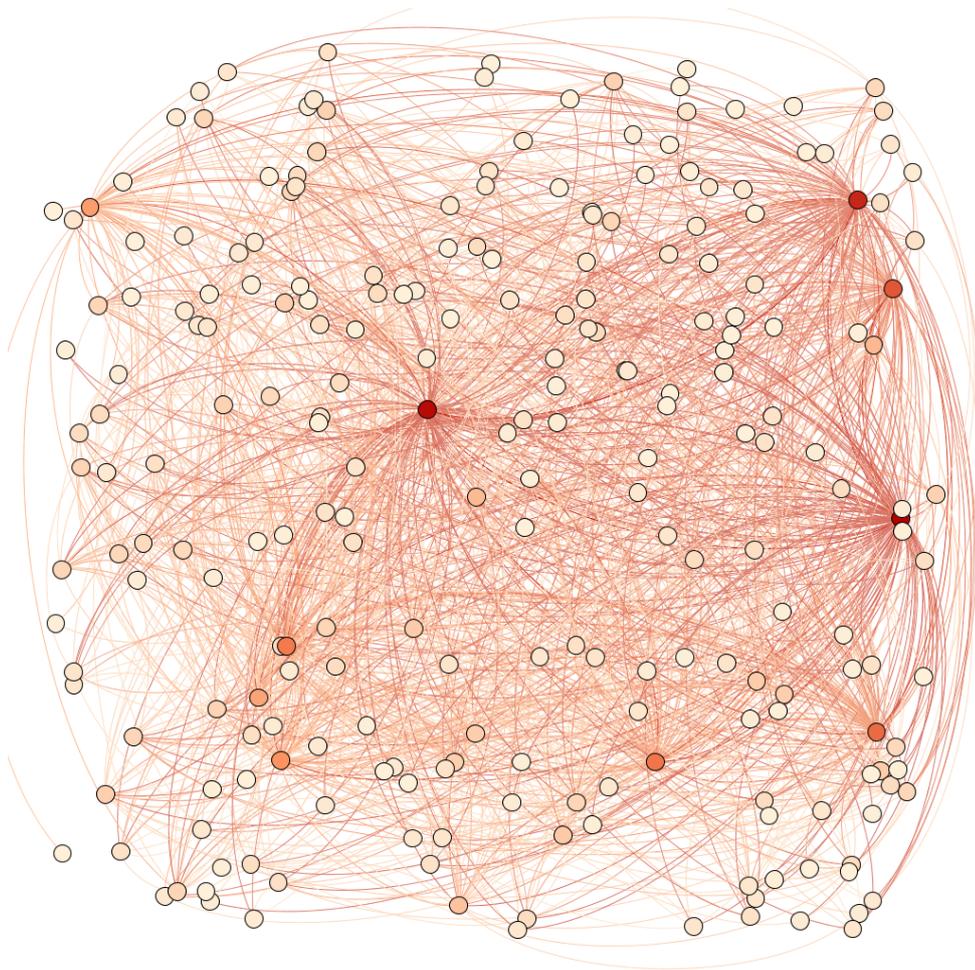
uma aresta, pois as arestas são colocadas aleatoriamente [19]. Este grafo é denominado  $G(N, p)$ , onde  $n$  representa o número de vértices e  $p$  a probabilidade de conexão de cada par de vértices do grafo. Neste modelo a probabilidade de um vértice possuir  $k$  arestas segue uma Distribuição de Poisson e é representada por:  $P(k) = e^{-\lambda} \lambda^k / k!$  onde  $\lambda = N(\frac{N-1}{k})p^k(1-p)^{n-1-k}$  [5].

Porém, em 1998, quando iniciou-se um projeto coordenado por Barabási com o objetivo de identificar o tamanho (diâmetro) da Web, juntamente com os estudos da tese de Watts relacionados ao fenômeno “Mundo Pequeno” ou “Small World” [52], acreditava-se que seria encontrada uma rede aleatória. No entanto, chegaram à seguinte conclusão: 80% das páginas tinham menos de 4 links, enquanto menos de 0,01% dos vértices tinha mais de 1000 links. A análise dos resultados mostrou que a distribuição seguia uma lei de potência e não uma distribuição de Poisson. Assim, observou-se que a web, embora aleatória, não seguia o padrão esperado, ou seja, a probabilidade de uma aresta conectar dois vértices não é a mesma para cada par de vértices [2].

Neste contexto, entra em destaque o estudo das redes complexas, que são grafos reais, como por exemplo, relacionamento entre pessoas, [36], redes sociais, empresas, organismos vivos [4], redes de energia [52], vírus e surto de doenças mortais [4], e até mesmo a internet [20, 6].

Tais redes reais têm características que são melhor modeladas por redes power law do que por modelos de grafos aleatórios, pois estas redes derivadas de dados reais são influenciadas por leis de potências [10, 6]. Na Figura 7 observa-se uma rede power law que foi utilizada nos experimentos (rede Airlines):

Como pode ser observado na Figura 7, alguns vértices possuem um enorme número de conexões com outros, enquanto que a maioria dos vértices possui poucas conexões. Os vértices mais populares (com maior número de conexões) são chamados de hubs. Estes vértices podem possuir centenas, milhares ou milhões de links [7].



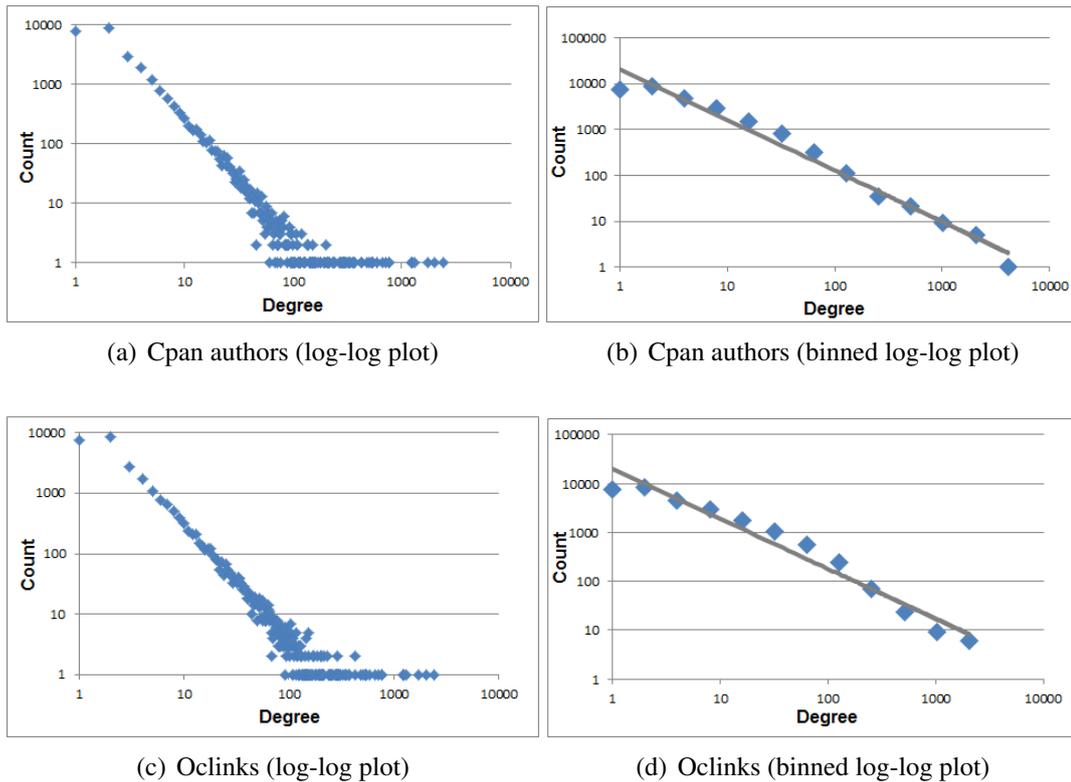
**Figura 7: Rede Power Law.**

### 2.5.1 Leis de Potência ou Power-Law

Nas últimas décadas do século XX, parte da comunidade dos físicos passou a se interessar pela dinâmica de sistemas complexos, cujas partes interagem de forma não-linear. Uma das propriedades marcantes de tais sistemas é a presença de leis de potência [10]

Leis de potência são relações matemáticas entre duas quantidades [25]. Diversas distribuições, representando tanto fenômenos naturais quanto humanos, que apresentam regularidades, são compostas por um grande número de fenômenos comuns e um pequeno número de fenômenos raros. Assim, a distribuição de conectividade das redes power law implica na minoria dos vértices com muitas ligações e uma abundância de vértices com poucas ligações. Historicamente, a primeira lei de potência descoberta foi o Princípio de Pareto [41] .

A distribuição de graus é uma das propriedades mais básicas quantitativas de uma rede, e pode ser definido pelo o número de arestas conectadas a um determinado vértice desta rede. Muitos vértices na Internet têm baixos graus de apenas uma ou duas ligações, mas alguns têm



**Figura 8: Distribuição de Graus.** O eixo  $x$  representa o valor do grau (de 0 até o grau máximo) e o eixo  $y$  representa o número de vértices com este grau.

um grau muito elevado, passando de mil ou mais em alguns casos. Em redes reais, a distribuição dos vértices possui uma longa cauda à direita. A Figura 8 exemplifica a distribuição de graus de dois grafos utilizados nos experimentos.

Existem dois mecanismos genéricos que fazem parte da maioria das redes artificiais e naturais: expansão contínua (adição de novos vértices) e uma certa preferência que os novos vértices da rede têm em se conectar com “hubs” (vértices de grau muito alto) já presentes na rede [5]. Redes que surgem seguindo estes princípios são conhecidas como redes de ligação preferencial (“preferential attachment”) e comumente são power law. Muitos sistemas físicos, químicos, biológicos e sociais podem ser interpretados como grafos onde os indivíduos são representados por vértices e suas interações por arestas. A maioria deles apresentam as características discutidas acima [6].

Redes power law existem em grandes quantidades na natureza e na sociedade. Seguem alguns exemplos de redes power law [7]:

**Tabela 2: Exemplos de redes de power law.**

Redes	Nós	Links
Metabolismo Celular	Moléculas envolvidas na queima de alimento para transformação em energia	Participação na mesma reação bioquímica
Hollywood	Atores	Aparecimento nos mesmos filmes
Internet	Roteadores	Conexão ótica e física
Rede de proteínas reguladoras	Proteínas que ajudam a regular as atividades celulares	Interações entre proteínas
Relações sexuais	Pessoas	Contato Sexual
WWW	Páginas da web	URLs

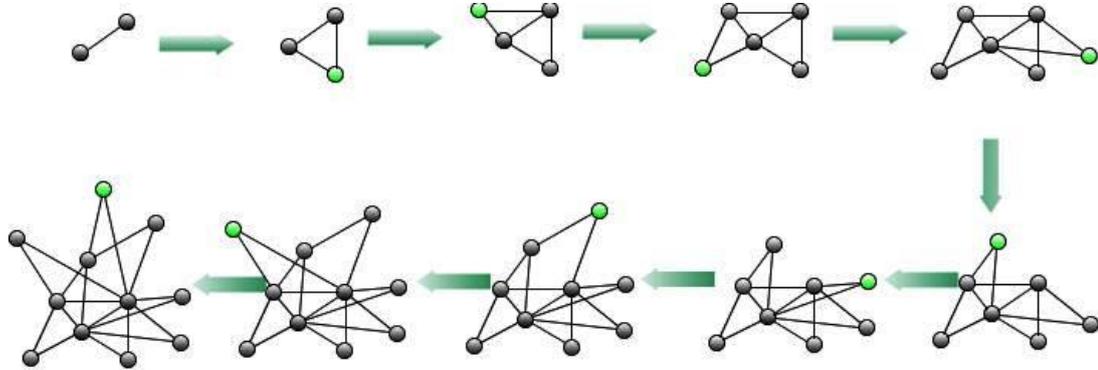
**Fonte: Adaptado de [7]**

Barabási e Bonabeau [7] citam algumas das principais aplicações de redes power law na área de computação, medicina e negócios:

- Computação:
  - Redes de computadores com a arquitetura power law (exemplo: a internet) são altamente resistentes a falhas acidentais;
  - Vírus da internet terão mais dificuldade em serem erradicados.
- Medicina:
  - Campanhas de vacinação poderiam ser mais efetivas se fossem concentradas no tratamento dos hubs.
  - Identificação de doenças nos hubs e o desenvolvimento de novas drogas.
- Negócios:
  - Monitoração de empresas interligadas, evitando catástrofes financeiras;
  - Estudo de novas formas de propagação de produtos no mercado.

A expansão da rede power law ocorre da seguinte forma (Figura 9 ): A rede inicia com um determinado conjunto de vértices, e progressivamente, novos vértices são adicionados e a rede

começa a ser expandida. Estes novos vértices decidem com quem se conectar e preferem os vértices que já têm mais conexões (conexão preferencial), ou seja, quanto maior o grau de um vértice, maior é a probabilidade que este vértice seja escolhido pelo novo vértice [7, 6].



**Figura 9: Processo de nascimento de uma rede power law: começando com um par de vértices adjacentes e em que cada novo vértice “escolhe” dois vértices da rede para se conectar.**

Fonte: Adaptado de [7]

### 3 COBERTURA POR VÉRTICES EM REDES COMPLEXAS

Uma cobertura em um grafo é um conjunto de vértices que toca todas as arestas do grafo. Uma cobertura mínima de vértices é o menor conjunto com tal propriedade. Este é um problema NP-Difícil clássico, que continua intratável até mesmo para grafos cúbicos e grafos planares com grau máximo 3. Portanto, algoritmos de aproximação para encontrar “boas” soluções em tempo polinomial são de grande interesse. Duas abordagens de aproximação mais comuns neste sentido são: Algoritmo Guloso e Algoritmo de 2-Aproximação:

- Algoritmo Guloso: Dado um grafo  $G$ , escolha o vértice  $v$  de maior grau e insira-o na cobertura  $S$ . Remova todos os vértices incidentes a  $v$ . Ordene os vértices por ordem de grau (do maior para o menor) e . Repita o processo iterativamente até que não haja mais arestas no grafo. Esta abordagem sempre encontra uma cobertura  $S$  tal que  $|S|$  é no máximo  $\ln n$  vezes o ótimo, onde  $n$  é o número de vértices de  $G$  [44].
- Algoritmo de 2-Aproximação: Dado um grafo  $G$ , escolha qualquer aresta  $uv$  e insira os vértices  $u$  e  $v$  na cobertura e remova  $u$  e  $v$  de  $G$ . Por último, remova as arestas incidentes a  $u$  e  $v$ . Repita a mesma estratégia em todo grafo iterativamente (as arestas sempre são escolhidas aleatoriamente), até que não haja mais arestas. Esta abordagem encontra uma cobertura de, no máximo, duas vezes o ótimo (ou seja, este é um algoritmo de aproximação com fator 2)[44].

O algoritmo de 2-aproximação apresentado acima é conhecido por cerca de 40 anos e é um dos mais utilizados até o momento para grafos em geral [14]. É conhecido que nenhum algoritmo pode garantir um fator melhor do que 1,36, a menos que  $P = NP$  [15]. De fato, conjectura-se uma afirmação ainda mais forte: para qualquer  $\varepsilon$  constante, não há uma  $(2 - \varepsilon)$ -aproximação para o problema [37]. Neste momento, o melhor algoritmo de aproximação conhecido, proposto por Karakostas [34] possui um fator de aproximação de  $2 - \Theta(\frac{1}{\sqrt{\log n}})$ , que é muito maior que os números que obtivemos experimentalmente usando a abordagem gulosa.

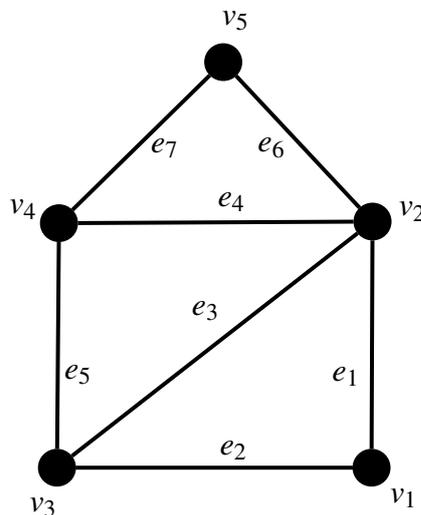
No domínio dos grafos power law, Gast e Hauptmann [29, 30, 31] propuseram recentemente um algoritmo e provaram que ele produz, em média, uma cobertura por vértices de 1.5 vezes o

ótimo. Este resultado foi o único que encontramos quando pesquisamos tópicos sobre cobertura por vértices em grafos power law.

Os algoritmos implementados nos experimentos se basearão na estratégia de aproximação. Na próxima seção serão realizadas execuções passo a passo com o algoritmo de 2-Aproximação e também com o algoritmo Guloso em um dado grafo.

### 3.1 EXECUÇÃO PASSO A PASSO DE ALGORITMOS PARA COBERTURA POR VÉRTICES

Considere o seguinte grafo  $G = (V, E)$ , tal que  $V = \{v_1, v_2, v_3, v_4, v_5\}$  e  $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$  representado graficamente pela figura 10.



**Figura 10: Grafo G com 5 vértices e 7 arestas.**

O grafo  $G$  é representado pela seguinte matriz  $5 \times 5$ :

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

A primeira execução será realizada com o algoritmo de 2-Aproximação.

### 3.1.1 Simulação com algoritmo de 2-Aproximação

Algoritmos de aproximação são utilizados para encontrar soluções aproximadas para problemas de otimização. Este tipo de algoritmo quase sempre está associado a problemas NP-Difícil.

Goodrich e Tamassia [27] descrevem um algoritmo de aproximação bastante simples, o qual envolve escolher uma aresta no grafo, colocar seus dois extremos na cobertura e apagar do grafo esta aresta e seus vértices incidentes. O resultado da execução é uma cobertura no máximo duas vezes maior que a ótima. Este é um algoritmo de aproximação de fator constante, no qual, o fator é igual a 2. O algoritmo baseia-se nos passos abaixo [12, 44]:

1. Encontre uma aresta descoberta
2. Adicione as suas extremidades na cobertura
3. Remova todas as arestas incidentes aos vértices do passo 2
4. Retorne ao passo 1 até que não existam mais arestas

---

#### **Algoritmo 1:** Algoritmo de 2-aproximação

---

**Entrada:** Grafo  $G = (V, E)$ , onde  $V$  representa o conjunto de vértices e  $E$  o conjunto de arestas

**Saída:** Cobertura por vértices  $C$

**início**

**repita**

        Escolha aleatoriamente uma aresta  $e(u,v)$  de  $G$  ;

        Coloque os vértices  $u$  e  $v$  em  $C$  ;

**para** cada aresta  $e$  incidente em  $u$  e  $v$  **faça**

            | Remova  $e$  de  $G$

**fim**

**até** Enquanto  $G$  contiver arestas;

    Retorne  $C$

**fim**

---

O algoritmo acima foi executado no grafo  $G$  (Figura 10) e resultou na seguinte cobertura  $C = \{v_1, v_2, v_3, v_4\}$  (Figura 11):

A figura 12 demonstra passo-a-passo a execução do algoritmo de 2-Aproximação:

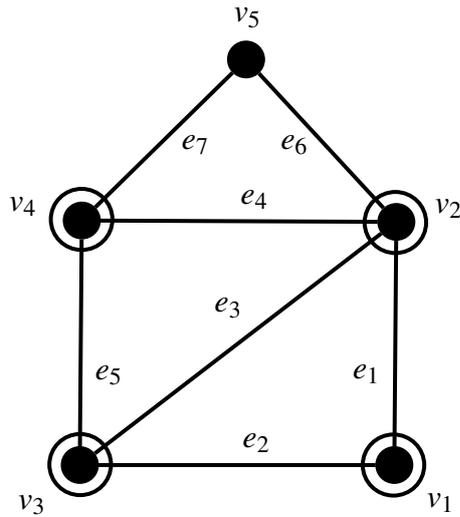
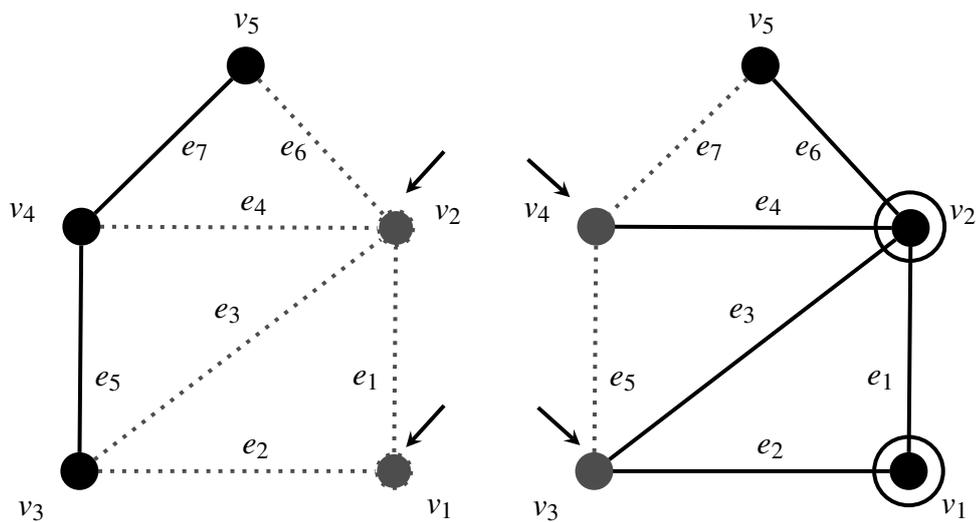


Figura 11: Resultado da Cobertura - Algoritmo de 2 Aproximação.  $C = \{v_1, v_2, v_3, v_4\}$



(a) Primeira iteração do Algoritmo de 2-Aproximação (b) Segunda iteração do Algoritmo de 2-Aproximação

**Figura 12: Execução do Algoritmo de 2-Aproximação. No primeiro passo, a aresta  $e_1\{v_1, v_2\}$  é escolhida. Todas as demais arestas incidentes aos vértices  $v_1$  e  $v_2$   $\{e_1, e_2, e_3, e_4, e_6\}$  são cobertas. No segundo passo, é escolhida a aresta  $e_5\{v_3, v_4\}$ , as arestas incidentes aos vértices  $v_3$  e  $v_4$   $\{e_5, e_7\}$  são cobertas. Os vértices  $v_1, v_2, v_3, v_4$  representam a cobertura por vértices do grafo**

Ao final da execução do algoritmo, o resultado é  $C = \{v_1, v_2, v_3, v_4\}$ .

Após a obtenção dos resultados acima, será realizada uma simulação com o algoritmo guloso.

### 3.1.2 Simulação com Algoritmo Guloso

Algoritmos para problemas de otimização normalmente passam por uma seqüência de etapas, com um conjunto de escolhas a cada passo. Um algoritmo guloso sempre faz a melhor escolha para aquele momento. Ou seja, ele faz uma escolha ótima local, esperando que isto leve à solução global ótima. Um algoritmo guloso jamais se “arrepende” ou volta atrás: as escolhas que faz em cada iteração são definitivas. Em cada passo, a escolha a ser feita deve ser [12, 44]:

- Possível, (satisfaz as restrições do problema);
- Localmente ótima, (deve ser a melhor escolha possível dentre as disponíveis neste passo);
- Irreversível, (não pode ser alterada nos passos subsequentes do algoritmo).

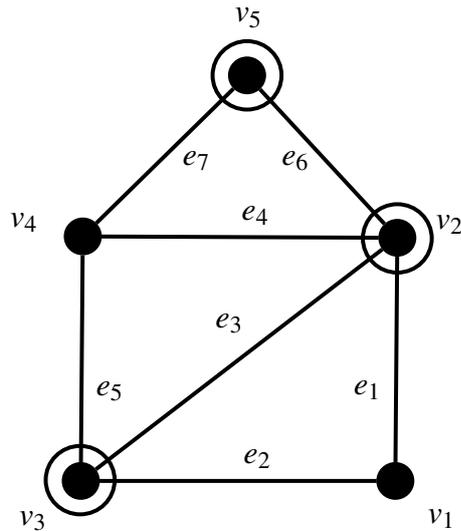
Exemplos de problemas que onde a estratégia gulosa encontra um resultado ótimo: problema de seleção de atividades, árvore geradora mínima, compressão de dados - Huffman, fórmula de Horn, entre outros[14].

Os elementos dos algoritmos gulosos estão listados à seguir:

1. Determinar a estrutura ótima do problema;
2. Desenvolver uma solução recursiva;
3. Provar que em cada estágio da recursão, uma das escolhas ótimas e a gulosa;
4. Desenvolver um algoritmo recursivo que implemente a estratégia gulosa;
5. Converter o algoritmo recursivo para iterativo.

Os passos do algoritmo guloso são os seguintes:

1. Encontre o vértice de maior grau;
2. Adicione-o na cobertura e remova-o de  $V$ ;
3. Remova todas as arestas incidentes a ele no grafo;
4. Retorne ao passo 1 até que não existam mais arestas.



**Figura 13: Resultado da Cobertura - Algoritmo Guloso.**  $C = \{v_2, v_3, v_5\}$

O algoritmo desenvolvido neste estudo baseia-se nos passos abaixo:

---

**Algoritmo 2:** Algoritmo Guloso

---

**Entrada:** Grafo  $G = (V, E)$ , onde  $V$  representa o conjunto de vértices e  $E$  o conjunto de arestas

**Saída:** Cobertura por vértices  $C$

**início**

$C \leftarrow 0$ ;

**repita**

Ordene  $V$  por ordem decrescente de grau;

**para cada vértice  $v$  de  $V$  faça**

**para cada aresta  $uw$  de  $E$  faça**

**se**  $((v == u)$  ou  $(v == w))$ , **então**

Remova  $uw$  da lista de arestas;

Atualize os graus de  $V$ ;

Adicione  $v$  em  $C$ ;

**fim**

**fim**

**fim**

**até** Enquanto  $G$  tiver arestas;

Retorne  $C$

**fim**

---

Uma execução gulosa poderia retornar a seguinte cobertura  $C = \{v_2, v_3, v_5\}$ . (Figura 13):

Abaixo, explica-se o passo-a-passo da execução do algoritmo de Guloso (Figura 14).

- Passo 1 - Os vértices são ordenados por grau. O primeiro vértice adicionado na cobertura é o  $v_2$ . As arestas adjacentes a ele  $e_1, e_3, e_4, e_6$  estão cobertas.

**Tabela 3: Grau dos vértices - Primeira Iteração.**

<b>vértice</b>	$v_2$	$v_4$	$v_3$	$v_5$	$v_1$
<b>Grau</b>	4	3	3	2	2

- Passo 2 - Na próxima iteração os graus são alterados e os vértices são ordenados novamente. O segundo vértice adicionado na cobertura é o  $v_3$ . As arestas adjacentes a ele  $e_5$  e  $e_2$  estão cobertas (A aresta  $e_3$  já havia sido coberta pelo  $v_2$ ).

**Tabela 4: Grau dos vértices - Segunda Iteração.**

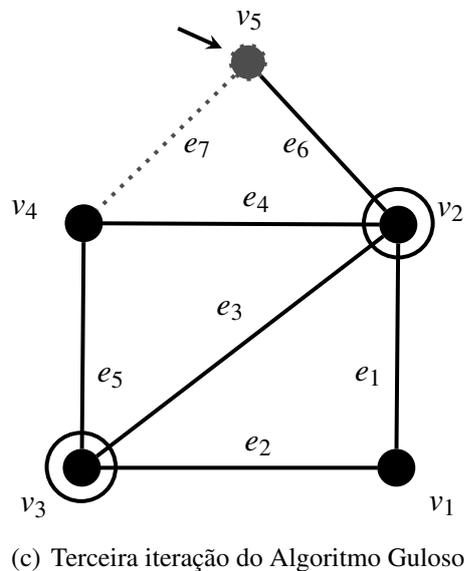
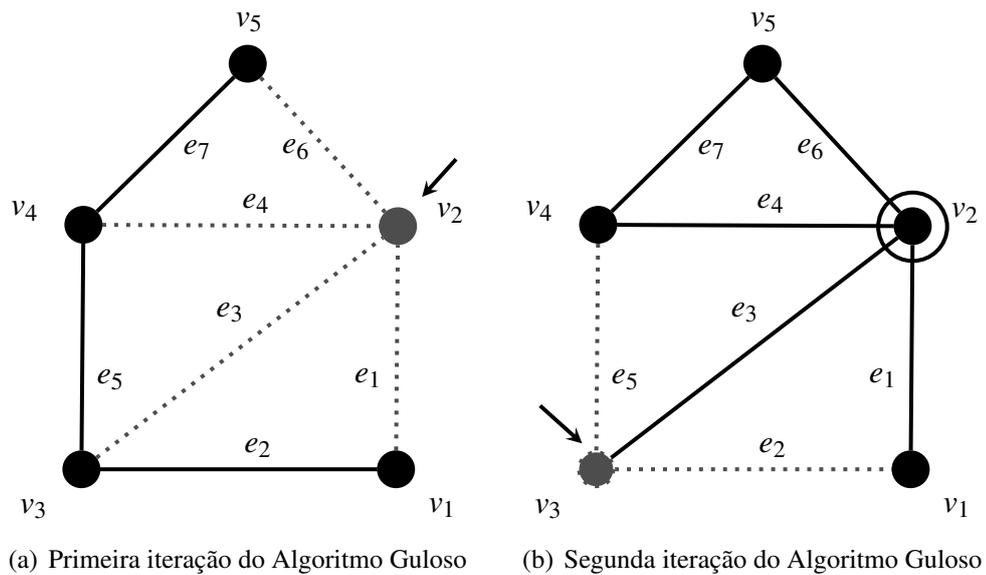
<b>vértice</b>	$v_4$	$v_3$	$v_5$	$v_1$	$v_2$
<b>Grau</b>	2	2	1	1	0

- Passo 3 - Na próxima iteração os graus são alterados e os vértices são ordenados novamente. O terceiro vértice adicionado na cobertura é o  $v_5$ . A única aresta restante, e incidente a ele,  $e_7$  está coberta (A aresta  $e_6$  já havia sido coberta pelo  $v_2$ ).

**Tabela 5: Grau dos vértices - Terceira Iteração.**

<b>vértice</b>	$v_4$	$v_5$	$v_1$	$v_2$	$v_3$
<b>Grau</b>	1	1	0	0	0

Após a execução da terceira iteração todas as arestas estão cobertas e todos os vértices tem grau zero.



**Figura 14: Execução do Algoritmo Guloso. No primeiro passo, o vértice  $v_2$  é escolhido. As arestas incidentes a ela  $e_1, e_3, e_4, e_6$  são cobertas. No segundo passo, é escolhida o vértice  $v_3$ , as arestas  $e_5$  e  $e_2$  são cobertas. Por fim, no terceiro passo o vértice  $v_5$  cobre a última aresta restante:  $e_7$ . Os vértices  $v_2, v_3, v_5$  representam a cobertura por vértices do grafo**

### 3.1.3 Cobertura de Vértices utilizando o Sage

O Sage (Sistemas de Experimentações Geométricas e Algébricas) é um software matemático que busca ser uma alternativa aos principais softwares matemáticos proprietários, tais como: Matlab, Magma, Maple e Mathematica [48]. Ele utiliza de pacotes matemáticos pré-existentes, como Maxima, GAP, Pari/GP, entre outros. Atualmente existem duas versões para utilização do Sage:

- Notebook: refere-se à uma interface acionada de dentro de um browser onde os passos são armazenados em páginas separadas por usuário. Exige o cadastro de um usuário e senha para utilização.
- Linha de comando: refere-se à versão instalada do software. É executada em uma máquina virtual (Virtual Box). Nesta versão é possível executar scripts em python escritos pelo usuário.

Existem diversos algoritmos relacionados à problemas em grafos implementados pelo Sage, tais como: cobertura por vértices, clique, coloração, conjunto independente, número cromático, entre outros. Em relação ao algoritmo relacionado ao problema de cobertura por vértices, o mesmo é baseado em uma técnica de redução ao núcleo do problema (*Kernalization*), que aplica 4 regras de redução. Sua implementação é baseada no algoritmo descrito em [1].

Neste estudo, foram realizados experimentos com grafos relativamente grandes (até 75 mil vértices), por isto, foi utilizada a versão instalada do Sage. Para cada grafo, foi gerado um script em python com os comandos para criação do grafo em memória (baseado na matriz de adjacência do mesmo) e execução do algoritmo de cobertura.

Os resultados obtidos pelo Sage representam o valor da cobertura “ótima” mencionada nos experimentos. Devido à natureza exponencial do algoritmo, não foi possível obter os resultados para todos os grafos utilizados nos experimentos.

Após o entendimento das principais características do Sage e das simulações apresentadas acima com o Algoritmo de 2-Aproximação e Guloso, serão demonstrados os resultados dos experimentos realizados com grafos reais.

## 4 RESULTADOS EXPERIMENTAIS

Nesta seção serão apresentados os resultados experimentais para o problema de cobertura mínima de vértices. Os algoritmos de 2-Aproximação e Guloso foram implementados em linguagem Java [17, 32]. O resultado denominado “Cobertura Ótima” foi obtido utilizando o software Sage [48].

### 4.0.4 Cenários das Simulações

Os experimentos foram executados em 25 grafos power law, extraídos de bases de grafos conhecidas [43, 49, 23, 24].

O tamanho dos grafos utilizados nos experimentos varia de algumas centenas até cerca de 75.000 vértices.

Abaixo, estão listados todos os grafos utilizados nos experimentos, acompanhados de uma sucinta descrição:

### 4.0.5 Airlines

Rede Tecnológica, possui 235 vértices, extraída de uma conhecida base de dados de grafos [23].

### 4.0.6 US Air

Rede Tecnológica, possui 332 vértices, extraída de uma conhecida base de dados de grafos [43].

### 4.0.7 Codeminer

Rede Social, possui 724 vértices, extraída de uma conhecida base de dados de grafos [23].

#### 4.0.8 Cpan authors

Rede Social, possui 839 vértices, extraída de uma conhecida base de dados de grafos [23]. Cpan é um projeto de visualização criado com o objetivo de analisar as relações entre os desenvolvedores e os pacotes da linguagem Perl, conhecido como Comunidade Cpan.

#### 4.0.9 EuroSis

Rede de Informação, possui 1285 vértices, extraída de uma conhecida base de dados de grafos [23]. Este grafo representa relações entre atores de doze países europeus.

#### 4.0.10 Oclinks

Rede de Informação, possui 1899 vértices, extraída de uma conhecida base de dados de grafos [23]

#### 4.0.11 YeastS

Rede Biológica, possui 2284 vértices, extraída de uma conhecida base de dados de grafos [43]. Representa interações entre proteínas.

#### 4.0.12 Grafo CA-GrQc

Rede de citação, possui 5242 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa colaborações científicas entre os autores de *papers* submetidos na categoria Relatividade Geral e Cosmologia Quântica do Arxiv [3]. Os dados abrangem os *papers* no período de janeiro de 1993 a abril de 2003 (124 meses).

#### 4.0.13 p2p-Gnutella08

Rede de comunicação, possui 6301, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 8 agosto de 2002.

#### 4.0.14 Wiki-Vote1

Rede Social, possui 7115 vértices, extraída de uma conhecida base de dados de grafos [49]. A rede contém todos os dados de votação para eleição de usuários administradores da Wikipedia desde o início do Wikipedia, até janeiro de 2008.

#### 4.0.15 p2p-Gnutella09

Rede de informação, possui 8114 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 9 agosto de 2002.

#### 4.0.16 p2p-Gnutella06

Rede de informação, possui 8717 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 9 agosto de 2002.

#### 4.0.17 p2p-Gnutella05

Rede de informação, possui 8846 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 5 agosto de 2002.

#### 4.0.18 CA-HepTh

Rede de citação possui 9875 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa colaborações científicas entre os autores de *papers* submetidos na categoria Fenomenologia do Arxiv [3]. Os dados abrangem os *papers* no período de janeiro de 1993 a abril de 2003 (124 meses).

#### 4.0.19 p2p-Gnutella04

Rede de colaboração, possui 10876 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 4 agosto de 2002.

#### 4.0.20 CA-AstroPh

Rede de colaboração possui 18771 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa colaborações científicas entre os autores de *papers* submetidos na categoria Astrofísica do Arxiv [3]. Os dados abrangem os *papers* no período de janeiro de 1993 a abril de 2003 (124 meses).

#### 4.0.21 p2p-Gnutella25

Rede de informação, possui 22687 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 25 agosto de 2002.

#### 4.0.22 CA-CondMat

Rede de colaboração possui 23133 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa colaborações científicas entre os autores de *papers* submetidos na categoria Matéria Condensada do Arxiv [3]. Os dados abrangem os *papers* no período de janeiro de 1993 a abril de 2003 (124 meses).

#### 4.0.23 p2p-Gnutella24

Rede de informação, possui 26518 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 24 agosto de 2002.

#### 4.0.24 Cit-HepTh

Rede de citação possui 23133 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa colaborações científicas entre os autores de *papers* submetidos na categoria Teoria da Física de Alta Energia do Arxiv [3]. Os dados abrangem os *papers* no período de janeiro de 1993 a abril de 2003 (124 meses).

#### 4.0.25 p2p-Gnutella30

Rede de informação, possui 27770 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento

de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 30 agosto de 2002.

#### 4.0.26 Email-Enron

Rede de comunicação possui 36692 vértices, extraída de uma conhecida base de dados de grafos [49]. Cobre toda a comunicação de e-mail dentro de um conjunto de dados de cerca de meio milhão de e-mails.

#### 4.0.27 Brightkite-edges

Rede social, possui 58228 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa um serviço de rede social baseado no *check-in* de usuários, onde são compartilhadas informações de sua localização.

#### 4.0.28 p2p-Gnutella31

Rede de informação, possui 62586 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma sequência de 9 *snapshots* da rede de compartilhamento de arquivos peer-to-peer Gnutella. Os dados abrangem o período de 30 agosto de 2002.

#### 4.0.29 soc-Epinions1

Rede social, possui 75879 vértices, extraída de uma conhecida base de dados de grafos [49]. Este grafo representa uma rede baseada em informações do site Epinions.com.

No próximo capítulo serão apresentados os resultados obtidos com experimentos realizados com base nos grafos descritos acima.

## 4.1 RESULTADOS

Para efeitos de comparação, também foi realizada a implementação do algoritmo de 2-Aproximação descrito neste capítulo e obtivemos resultados piores do que o algoritmo Guloso. Uma típica cobertura por vértices gerada a partir do algoritmo de 2-Aproximação é, em média, 1.5 vezes a ótima. Estes resultados foram obtidos com base nos 15 grafos nos quais foi possível encontrar o resultado ótimo para comparação. Estes resultados estão apresentados nas tabelas abaixo.

Em 15 destas redes nossos experimentos mostram que uma estratégia gulosa gera um resultado em média 1.02 vezes o ótimo. O valor das coberturas encontradas são descritos na tabela 7 e a taxa de aproximação na coluna denominada Guloso/Ótimo na tabela 8.

A fim de obter o valor ótimo para a cobertura por vértices, utilizou-se o uma biblioteca chamada SAGE [48]. Os experimentos foram executados em um Servidor DELL (Intel Xenon Quad Core 5500-5600, com 60GB de memória swap + 12GB de memória física.) Conseguimos obter coberturas mínimas para grafos com mais de de 10000 vértices, mas para os 10 maiores grafos que não foi possível chegar a um resultado devido à natureza exponencial do algoritmo. Mesmo assim, é um pouco surpreendente que tenha sido possível computar o valor ótimo para grafos com mais de 10000 vértices.

Foi possível obter resultados para todos os grafos executando os algoritmos de 2-Aproximação e o algoritmo guloso, porém não conseguimos a cobertura ótima (baseada no Sage [48]) para todos os grafos.

Os resultados apontam que o algoritmo implementado com base na estratégia gulosa obteve uma cobertura de tamanho 0.66 vezes o tamanho da cobertura do algoritmo concorrente (2-Aproximação). Estes resultados estão presentes na coluna Guloso/2-Aproximação da tabela.

Em uma nota final, como já mencionamos antes, em nossos experimentos também foi calculada a relação entre o tamanho da cobertura obtida pelo algoritmo Guloso dividida pelo tamanho da cobertura obtida pelo algoritmo de 2-Aproximação e os resultados mostram que a cobertura obtida pelo algoritmo Guloso é de cerca de 66% do tamanho da outra abordagem. Deve também ser salientado que este número parece ser invariante ao tamanho do gráfico. Este fato pode nos dar alguma indicação de que mesmo para as 10 maiores gráficos (para os quais os valores ótimos não estavam disponíveis para efeitos de comparação), o tamanho da cobertura obtido pelo algoritmo Guloso pode estar mais perto ainda do ótimo.

**Tabela 6: Redes Reais utilizadas nos experimentos.**

ID	Nome do Grafo	# Vértces	Arestas	Tipo do Grafo	Base
1	Airlines	235	1295	Tecnológico	Gephi
2	US Air	332	2126	Tecnológico	Pajek
3	Codeminer	724	1015	Rede Social	Gephi
4	Cpan authors	839	2112	Rede Social	Gephi
5	EuroSis	1285	6462	Informação	Gephi
6	Oclinks	1899	13821	Informação	Gephi
7	YeastS	2284	6646	Biológico	Pajek
8	CA-GrQc	5241	14484	Citação	Stanford
9	p2p-Gnutella08	6301	20776	Informação	Stanford
10	Wiki-Vote1	7115	100729	Rede Social	Stanford
11	p2p-Gnutella09	8114	26013	Informação	Stanford
12	p2p-Gnutella06	8717	31523	Informação	Stanford
13	p2p-Gnutella05	8846	31837	Informação	Stanford
14	CA-HepTh	9875	25973	Citação	Stanford
15	p2p-Gnutella04	10876	39993	Informação	Stanford
16	CA-AstroPh	18771	198050	Colaboração	Stanford
17	p2p-Gnutella25	22687	54705	Informação	Stanford
18	CA-CondMat	23133	93439	Colaboração	Stanford
19	p2p-Gnutella24	26518	65368	Informação	Stanford
20	Cit-HepTh	27769	352285	Citação	Stanford
21	p2p-Gnutella30	36682	88328	Informação	Stanford
22	Email-Enron	36692	183811	Comunicação	Stanford
23	Brightkite-edges	58228	214023	Rede Social	Stanford
24	p2p-Gnutella31	62586	147890	Informação	Stanford
25	soc-Epinions1	75879	404953	Rede Social	Stanford

**Tabela 7: Resultados dos experimentos.**

ID	Nome do Grafo	# Vértces	"Ótimo"	Guloso	2-Aprox
1	Airlines	235	96	97	146
2	US Air	332	149	151	230
3	Codeminer	724	191	196	334
4	Cpan authors	839	116	117	196
5	EuroSis	1285	597	608	896
6	Oclinks	1899	749	763	1100
7	YeastS	2284	763	773	1240
8	CA-GrQc	5241	2783	2795	3960
9	p2p-Gnutella08	6301	2054	2070	3366
10	Wiki-Vote1	7115	2249	2370	3460
11	p2p-Gnutella09	8114	2574	2589	4238
12	p2p-Gnutella06	8717	3405	3484	5352
13	p2p-Gnutella05	8846	3428	3475	5412
14	CA-HepTh	9875	4981	5003	7240
15	p2p-Gnutella04	10876	4348	4428	6624
16	CA-AstroPh	18771	n.a	12044	15194
17	p2p-Gnutella25	22687	n.a	6055	9800
18	CA-CondMat	23133	n.a	13561	18150
19	p2p-Gnutella24	26518	n.a	7250	11624
20	Cit-HepTh	27769	n.a	18225	23396
21	p2p-Gnutella30	36682	n.a	9321	15096
22	Email-Enron	36692	n.a	14477	20674
23	Brightkite-edges	58228	n.a	22177	34814
24	p2p-Gnutella31	62586	n.a	15864	25582
25	soc-Epinions1	75879	n.a	22418	35964

**Tabela 8: Resultados dos experimentos**

ID	Guloso /Ótimo	2-App /Ótimo	Guloso /2-App
1	1,01	1,52	0,66
2	1,01	1,54	0,66
3	1,03	1,75	0,59
4	1,01	1,69	0,60
5	1,02	1,50	0,68
6	1,02	1,47	0,69
7	1,01	1,63	0,62
8	1,00	1,42	0,71
9	1,01	1,64	0,61
10	1,05	1,54	0,68
11	1,01	1,65	0,61
12	1,02	1,57	0,65
13	1,01	1,58	0,64
14	1,00	1,45	0,69
15	1,02	1,52	0,67
16	n.a	n.a	0,79
17	n.a	n.a	0,62
18	n.a	n.a	0,75
19	n.a	n.a	0,62
20	n.a	n.a	0,78
21	n.a	n.a	0,62
22	n.a	n.a	0,70
23	n.a	n.a	0,64
24	n.a	n.a	0,62
25	n.a	n.a	0,62
<b>Média</b>	<b>1,02</b>	<b>1,56</b>	<b>0,66</b>
<b>Desvio Padrão.</b>	<b>0,01</b>	<b>0,09</b>	<b>0,05</b>

No gráfico da Figura 15 abaixo, fica claro como o resultado do algoritmo guloso está realmente muito próximo do ótimo (as linhas "Ótimo" e "Guloso" se sobrepõem):

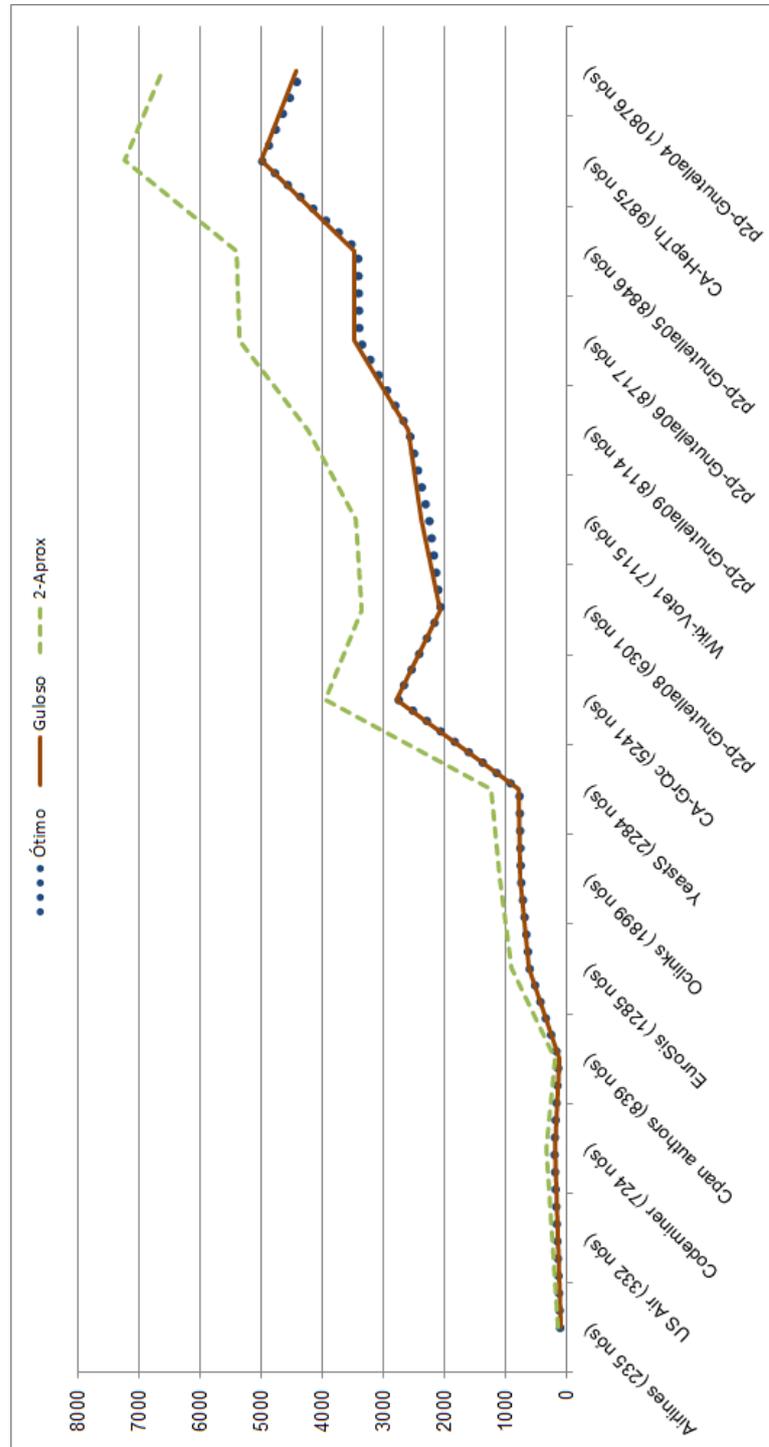


Figura 15: Gráfico com os resultados dos 15 primeiros grafos analisados.

## 5 CONCLUSÃO

Neste trabalho foi realizado um estudo experimental em relação ao comportamento de um algoritmo baseado em uma estratégia gulosa para o problema de cobertura por vértices e o resultado deste estudo foi comparado com outro algoritmo de aproximação e com a solução exponencial ótima. Em particular esta solução foi aplicada e analisada em redes complexas. Foram realizados experimentos que demonstram quem, embora a estratégia gulosa não garanta teoricamente uma “boa” aproximação, ela obtém melhores resultados que o algoritmo de 2-Aproximação.

Uma cobertura em um grafo é um conjunto de vértices cuja remoção desconecta totalmente o grafo. Uma cobertura mínima de vértices é o menor conjunto com tal propriedade. Este é um problema NP-Difícil clássico. Redes power law são redes complexas cujo grau de distribuição segue leis de potência. Estas redes possuem poucos “hubs”(vértices de grau muito elevado), naturalmente, pode-se esperar que uma abordagem gulosa (escolhendo primeiro estes hubs para a cobertura) terá uma boa performance. A idéia é que o algoritmo guloso explore esta característica das redes power law.

Após a apresentação do algoritmo, o mesmo foi implementado em Java para simulações. Foram realizadas simulações com 25 grafos reais. Além dos testes com o algoritmo guloso, foram realizados experimentos com o algoritmo de 2-Aproximação e estes dois resultados foram comparados experimentalmente com o resultado da “cobertura ótima” obtida pela biblioteca Sage (software matemático). Em virtude da natureza exponencial do algoritmo implementado pelo Sage, não foi possível obter o resultado da cobertura para os 10 maiores grafos da tabela 6.

Nas tabelas 7 e tabela 8 podemos analisar o resultado da cobertura para os 25 grafos reais utilizados nos experimentos. O tamanho dos grafos utilizados nos experimentos varia de algumas centenas até cerca de 75.000 vértices. Em 15 destas redes nossos experimentos mostram que uma estratégia gulosa gera um resultado em média 1.02 vezes o ótimo.

Também implementamos o algoritmo de 2-aproximação e foram obtidos resultados significativamente piores que o algoritmo guloso. A cobertura por vértices obtida pelo algoritmo de

2-aproximação foi de cerca de 1.56 vezes o ótimo (estes resultados foram obtidos a partir dos 15 grafos dos quais possuímos o resultado ótimo para comparação).

Os experimentos foram realizados tanto com o algoritmo guloso quanto com o algoritmo de 2-Aproximação nos 25 grafos (isto é, não apenas nos 15 grafos em que conseguimos o valor ótimo para comparação) e os resultados indicaram que a abordagem gulosa obtém uma cobertura por vértices de tamanho 0.66 vezes o tamanho do algoritmo concorrente, ou seja, a cobertura obtida pelo algoritmo guloso é 66 % do tamanho da 2-Aproximação. É importante salientar também que este número parece ser invariante ao tamanho do grafo. Este fato nos indica que, até mesmo para os 10 maiores grafos (para os quais os valores da cobertura ótima não estavam disponíveis para comparação, o tamanho das coberturas obtidas pelo algoritmo guloso parecem estar satisfatoriamente muito perto do valor ótimo.

Por fim, gostaríamos de apresentar algumas questões que ainda estão em aberto (tabela 9) no contexto deste trabalho e merecem um estudo mais aprofundado:

- Qual é a garantia teórica que o Algoritmo de Hauptmann obtém em grafos em geral para o problema de cobertura?
- Qual é a garantia probabilística (teórica) para o algoritmo de 2-Aproximação e para o algoritmo Guloso em grafos power law para o problema de cobertura?
- Na prática, como o Algoritmo de Hauptmann se comporta em grafos reais?

**Tabela 9: Trabalhos Futuros.**

	2-Aprox	Guloso	Hauptmann [29]
<b>Grafos em Geral (Teórico)</b>	2	$\ln n$	?
<b>Power Law (Teórico)</b>	?	?	1.5
<b>Grafos Reais</b>	1.56	1.02	?

## REFERÊNCIAS

- [1] ABU-KHZAM, F. N.; COLLINS, R. L.; FELLOWS, M. R.; LANGSTON, M. A.; SUSTERS, W. H.; SYMONS, C. T. *Kernelization Algorithm for the Vertex Cover Problem: Theory and Experiments*. SIAM ALENEX/ANALCO, p. 62-69, 2004.
- [2] ALBERT, R.; JEONG, H.; BARABÁSI, A. L. *Diameter of the World-Wide Web* Nature, v. 401, p. 130-131, 1999.
- [3] ARXIV, <http://arxiv.org>, acesso em 02/02/2012.
- [4] BARABASI, A.L. *Linked: How Everything Is Connected to Everything Else and What It Means*. New York: Plume, 2003.
- [5] BARABÁSI, A-L.; ALBERT, R. *Emergence of scaling in random networks*. Science, v. 286, p. 509-512, 1999.
- [6] BARABÁSI, A-L.; ALBERT, R. *Statistical Mechanics of Complex Networks*. Reviews of Modern Physics, v.74, p. 47-97, 2002.
- [7] BARABÁSI, A-L; BONABEAU, E. *Scale-Free Networks*. Scientific American v. 288, p. 50-59, 2003.
- [8] BOLLOBÁS, B.; RIORDAN, O.; SPENCER, J.; TUSNDY, G. *The Degree Sequence of a Scale-Free Random Graph Process*. Random Structures & Algorithms. Vol. 18 Issue 3, p. 279-290, 2001.
- [9] BRODER, A.; KUMAR, R.; MAGHOUL, F.; RAGHAVAN, P.; RAJAGOPALAN, S; STATA, R.; TOMKINS, A. S.; WIENER, J. *Graph structure in the Web*. Computer networks, 33(1-6):309-320, 2000.
- [10] CLAUSET, A.; SHALIZI, C. R.; NEWMAN, M. E. J. *Power-law distributions in empirical data*. SIAM Review, v. 51, p. 661-703, 2009.
- [11] CHUNG, F; LU, L. *Complex Graphs and Networks*. American Math. Society, 2006.
- [12] CORMEN, T.H et al. *Introduction to Algorithms*. 3<sup>a</sup> ed: USA.MIT Press and Mac Graw-Hill, 2009.
- [13] COSTA, L. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. VI. *Characterization of complex networks: A Survey of measurements*. Advances in Physics, v. 56, p. 167-242, 2005.
- [14] DASGUPTA, S.; PAPADIMITRIOU, C.H.; VAZIRANI, U.V. *Algorithms*. MG-Hill, 2008.
- [15] DINUR, I.; SAFRA, S. *On the hardness of approximating vertex cover*. Annals of Mathematics, p. 439-485, v. 162, 2005.

- [16] DEVLIN, Keith. *Os Problemas do Milênio: Sete grandes enigmas matemáticos*. São Paulo: Record, 2004.
- [17] ECLIPSE Juno, <http://www.eclipse.org/juno/>, acessado em 20/10/2012.
- [18] EUBANK, S.; KUMAR, V. S. A.; MARATHE, M. V.; Srinivasan, A.; WANG, N. *Structural and algorithmic aspects of massive social networks*. In Proceedings of the 15th annual ACM-SIAM Symposium on Discrete Algorithms, pages 718-727. SIAM, 2004.
- [19] ERDOS, P.; RENYI, A. *On random graphs*. Publicationes Mathematicae (Debrecen), v. 6, p. 290-297, 1959.
- [20] FALOUTSOS, M.; FALOUTSOS, P.; FALOUTSOS, C. *On power-law relationships of the Internet topology*. In: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, v. 29 of SIGCOMM '99, 1999.
- [21] GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co. New York, NY (1979).
- [22] GKANTSIDIS, C.; MIHAIL, M.; SABERI, A. *Conductance and congestion in power law graphs*. SIGMETRICS Performance Evaluation Review, 31:148-159, 2003.
- [23] GEPHI Datasets, <http://wiki.gephi.org/index.php/Datasets>, acesso em 01/02/2012
- [24] GEPHI Marketplace, <https://marketplace.gephi.org/>, acesso em 01/02/2012
- [25] GLERIA, I.; MATSUSHITA, R.; SILVA, S. *Sistemas complexos, criticalidade e leis de potencia*. Revista Brasileira de Ensino de Física, v. 26, n. 2, p. 99 - 108, 2004.
- [26] GOLD, S.; RANGARAJAN, A. *A Graduated Assignment Algorithm for Graph Matching*. IEEE Transactions on Pattern Analysis and Machine Intelligence, v.18, n.5, p.377-388, 1996.
- [27] GOODRICH, M. T.; TAMASSIA, R. *Projeto de Algoritmos*. São Paulo: Bookman Companhia Ed. 2004.
- [28] GUELZIM, N.; Bottani, S.; BOURGINE, P.; KÉPÈS, F. *Topological and causal structure of the yeast transcriptional regulatory network*. Nature Genetics, 31(1):60-63, 2002.
- [29] GAST M.; HAUPTMANN, M. *Approximability of the Vertex Cover Problem in Power Law Graphs*. Computing Research Repository (CoRR), arXiv:1204.0982, 2012.
- [30] HAUPTMANN, M.; GAST M.; KARPINSKI, M. *Inapproximability of Dominating Set in Power Law Graphs*. Computing Research Repository (CoRR), arXiv:1212.3517, 2012.
- [31] HAUPTMANN, M.; GAST M.; KARPINSKI, M. *Improved Approximation Lower Bounds for Vertex Cover on Power Law Graphs and Some Generalizations*. Computing Research Repository (CoRR), arXiv:1210.2698, 2012.
- [32] Java, <http://www.oracle.com/br/technologies/java/overview/index.html>, acessado em 20/10/2012.

- [33] JOVANOVIĆ, M.; ANNEXSTEIN, F. S.; BERMAN, K. A. *Modeling peer-to-peer network topologies through “small-world” models and power laws*. In IX Telecommunications Forum, TELFOR, 2001
- [34] KARAKOSTAS, G. *A better approximation ratio for the vertex cover problem*. In: ACM Transactions on Algorithms. V 5, Issue 4, out, 1999.
- [35] KARP, R. M. *Reducibility Among Combinatorial Problems*. The Journal of Symbolic Logic. v. 40, p. 618-636, dez. 1975.
- [36] LESKOVEC, J.; HUTTENLOCHER, D. KLEINBERG, J. *Signed networks in social media*. In: Proceedings of 28th international conference on Human factors in computing systems (CHI '10). ACM, p. 1361-1370, 2010.
- [37] KHOT, S; REGEV, O. *Vertex cover might be hard to approximate to within  $2 - \epsilon$* . Journal of Computer and System Sciences Volume 74, Issue 3, p. 335-349, May 2008.
- [38] EASLEY, D.; KLEINBERG, J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press (2010)
- [39] KLEINBERG, J.; KUMAR, R.; RAGHAVAN, P.; RAJAGOPALAN, S.; TOMKINS, A. S. *The Web as a graph: measurements, models and methods*. Proceedings of the 5th annual international Conference on Computing and Combinatorics, 1-17, 1999.
- [40] MIHAIL, M.; PAPADIMITRIOU, C. H.; SABERI, A. *On certain connectivity properties of the internet topology*. Journal of Computer and System Sciences, 72(2):239-251, 2006.
- [41] NEWMAN, M. *Power laws, Pareto distributions and Zipf's law*. Contemporary Physics 46, 323–351. 2005.
- [42] NEWMAN, M. *The physics of networks*. In: Physics Today, p 33–38, Novembro 2008.
- [43] PAJEK Datasets, <http://vlado.fmf.uni-lj.si/pub/networks/data>
- [44] PAPADIMITRIOU, C. H.; STEIGLITZ, K. *Combinatorial optimization: algorithms and complexity*. Prentice Hall (1982).
- [45] PARK, K.; LEE, H. *On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets*. ACM SIGCOMM Computer Communication Review, 31(4):15-26, October 2001.
- [46] PAPOULIS, A. *Poisson Process and Shot Noise*. Ch. 16 in Probability, Random Variables, and Stochastic Processes, 2nd ed. New York: McGraw-Hill, pp. 554-576, 1984.
- [47] RABUSKE, M. A. *Introdução à Teoria dos Grafos*. Florianópolis: UFSC, 1992. 184 p.
- [48] SAGE, <http://www.sagemath.org/library.html>, acessado em 02/03/2012.
- [49] Stanford Large Network Dataset Collec. <http://snap.stanford.edu/data>, acessado em 02/03/2012.
- [50] STEGE, U., R.; FELLOWS. *An improved fixed-parameter-tractable algorithm for Vertex Cover*. Technical Report 318, Department of Computer Science, ETH Zurich, Abril 1999.

- [51] SZWARCFITER, Jayme Luiz. *Grafos e algoritmos computacionais*. Rio de Janeiro: Campus, 1988. 223 p.
- [52] WATTS, D. J.; STROGATZ, S. H. *Collective dynamics of small work networks*. Nature, v.393, p 440-442, Junho 1998.
- [53] WEST, Douglas B. *Graph Theory*. 2 ed. New York: Prentice Hall, 2002. 470 p.
- [54] WILSON R. J. *An Eulerian trail through Konigsberg*. Journal of Graph Theory, v.10, p. 265-275, 1986.